

# Desarrollo de una plataforma de Cloud Manufacturing

Informe Final de Proyecto de Grado  
Ingeniería en Computación

Santiago Chiappa  
Emiliano Videla

Tutores:

Pedro Piñeyro  
Daniel Rossit  
Víctor Viana

Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República  
Uruguay

Febrero de 2022





## Resumen

La creciente incorporación de tecnologías digitales en los sistemas industriales en los últimos años y su correspondiente conexión con *Internet of Things*, ha permitido ofrecer servicios de producción a través de la nube. Esto ha dado lugar al paradigma de producción de *Cloud Manufacturing*, que permite a clientes y proveedores confluír en una plataforma virtual donde generar oportunidades de negocio.

En este trabajo se presenta una investigación exhaustiva del estado del arte de *Cloud Manufacturing*, el nuevo paradigma de fabricación en la nube apoyado fuertemente en el uso de *Internet of Things* que permite en gran medida, optimizar los recursos y reducir costos. *Cloud Manufacturing* permite que diferentes proveedores de servicios puedan asociarse entre sí para mejorar el nivel de servicio, así como para permitir que otro proveedor aproveche su capacidad ociosa.

Durante la revisión se encontró que los temas más recurrentes son virtualización, compartición de recursos, planificación, almacenamiento de datos, supervisión, análisis de datos, seguridad y optimizaciones. Adicionalmente, se realizó un estudio de Eclipse BaSyx, una potente plataforma cuyo fin es brindar un conjunto de componentes comunes y reusables de la industria 4.0 que permitan el rápido desarrollo de soluciones de industria 4.0. Para ello provee un conjunto de clases exportadas para distintos lenguajes mediante kits de desarrollo de software. Estas clases permiten la creación de *Digital Twins* para distintos tipos de activos, así como su integración y comunicación, permitiendo así la virtualización de distintos dispositivos físicos.

Finalmente, se realizó la implementación de un prototipo de sistema de fabricación en la nube con el objetivo de crear una prueba de concepto para validar los conceptos y decisiones arquitectónicas de *Cloud Manufacturing* y validar los componentes ofrecidos por Eclipse BaSyx.

Sobre este sistema se realizaron diferentes experimentos: manuales usando la interfaz web para evaluar el sistema a nivel funcional y automatizadas para evaluar el desempeño del sistema y la influencia del módulo planificador.

De las pruebas se desprende que se logró desarrollar, usando BaSyx, un sistema funcional que satisface los requerimientos mínimos de un sistema de CMfg, mediante el cual es posible trasladar conceptos comunes de fabricación a un sistema de fabricación en la nube, poniendo el foco en la fabricación de lote de tamaño 1 y brindando soporte al monitoreo de los procesos y los dispositivos que intervienen en él.

**Palabras claves:** Cloud Manufacturing, Arquitecturas de software, Internet of Things, Sistemas Ciber-Físicos, Virtualización, Planificación, Industria 4.0, *Digital Twins*



## Trabajos relacionados

El desarrollo de este proyecto de grado ha dado lugar a los siguientes trabajos relacionados:

- **Congreso XXXIII ENDIO - XXXI EPIO RED-M IX, 2020**
  - **Título:** Estado del arte de Cloud Manufacturing
  - **Autores:** Santiago Chiappa, Emiliano Videla, Pedro Piñeyro, Victor Viana, Daniel Rossit
  - **Lugar:** Córdoba, Argentina
  - **Año:** 2020
  - **Estado:** Trabajo publicado
- **Journal of Industrial Information Integration, Elsevier BV**
  - **Título:** Cloud Manufacturing architectures: state-of-art, research challenges and platforms description
  - **Autores:** Santiago Chiappa, Emiliano Videla, Victor Viana, Pedro Piñeyro, Daniel Rossit
  - **Lugar:** Países Bajos
  - **Año:** 2021
  - **Estado:** Trabajo enviado

## Agradecimientos

En primera instancia, agradecer a Pedro Piñeyro, Daniel Rossit y Victor Viana, los tutores de esta tesis de grado quienes, con mucha dedicación y paciencia, hicieron que el camino recorrido durante el tiempo que duró este trabajo fuese sumamente agradable y motivador. Agradecerles su apoyo, sus consejos y las correcciones que fueron realizando durante este proceso.

En segundo lugar, agradecer a nuestras familias y amigos, que siempre nos transmitieron su apoyo incondicional durante todos los años de carrera, y donde el desarrollo de esta tesis no fue la excepción. La motivación que transmitieron siempre fue fundamental para que hoy estemos donde estamos.

Finalmente, agradecer a todos los investigadores y autores que generan y publican conocimiento de forma libre; artículos científicos que permiten el avance del conocimiento y hacen que las tareas de investigación sean más accesibles.

A todos, ¡muchas gracias!



# Tabla de contenido

1. Introducción	<b>1</b>
1.1 Alcance	1
1.2 Objetivos	2
1.3 Resultados	2
1.4 Estructura del documento	2
2. Estado del arte	<b>4</b>
2.1 Contexto	4
2.2 Industria 4.0	4
2.3 Beneficios de Cloud Manufacturing	6
2.4 Características	7
3. Eclipse BaSyx	<b>23</b>
3.1 Por qué usar Eclipse BaSyx en un sistema de CMfg	24
3.2 Componentes principales	25
3.3 Utilizando BaSyx en un sistema de CMfg	28
4. Arquitectura	<b>29</b>
4.1 Planificador o Scheduler	31
4.2 Verificador o Monitoring	31
4.3 Eclipse BaSyx	33
5. Implementación	<b>34</b>
5.1 Aplicación web	34
5.2 Digital Twins de BaSyx	40
5.3 Base de datos	44
5.4 Diseño experimental	44
6. Experimentación	<b>47</b>
6.1 Casos de prueba	48
6.2 Resultados obtenidos	50
6.3 Análisis y comparaciones	53
7. Conclusiones y trabajos a futuro	<b>54</b>
Glosario	<b>56</b>
Bibliografía	<b>58</b>
<b>Anexos</b>	<b>63</b>



# 1. Introducción

El surgimiento del concepto de Industria 4.0 se debe al gran avance tecnológico que ha tenido lugar en las últimas décadas. De este avance tecnológico y del proceso de globalización se desprende un aumento en la competitividad y en las exigencias del consumidor. Los consumidores tienen requerimientos cada vez más específicos. Con el fin de resolver estos requerimientos, comienza a surgir el concepto de “lote de tamaño 1” [12, 65]. *Cloud Manufacturing* (CMfg) es un paradigma que surge con el fin de aprovechar los grandes avances tecnológicos y las tecnologías IoT, para así poder solucionar estas nuevas necesidades emergentes.

Este concepto consiste, fundamentalmente, en conectar los procesos de fabricación con la nube de forma tal que permita que las empresas cooperen y funcionen virtualmente como si fuesen una única gran empresa en la nube.

Si bien no existe una definición precisa de CMfg [2, 7, 29], a medida que la tecnología avanza, el concepto de CMfg también lo hace, hasta el punto de que hoy en día pueden encontrarse muchas investigaciones y menciones sobre este.

Por este motivo, resultó muy interesante realizar un aporte más orientado al área de programación.

El objetivo planteado es lograr determinar con mayor exactitud el estado actual de CMfg y las distintas herramientas disponibles para ponerlo en práctica. De esta forma, un segundo objetivo consiste en poder implementar un prototipo que permita mostrar las virtudes y defectos de un sistema de CMg aplicado a un ejemplo real.

## 1.1 Alcance

El trabajo llevado a cabo consta de dos etapas bien diferenciadas.

En primer lugar, se decidió realizar una revisión sistemática de literatura para estudiar el estado del arte de CMfg. Para ello fueron relevados 174 artículos científicos de las bases de datos provistas por los buscadores de Scopus, IEEE Xplore y Springer Link. El detalle del proceso de revisión sistemática de literatura puede leerse en profundidad en el Anexo I. Sus resultados se encuentran resumidos en la Sección 2 de este documento.

En segundo lugar, realizar la implementación de un prototipo de un sistema de CMfg, a modo de prueba de concepto, aprovechando este desarrollo para validar el *framework*<sup>1</sup> de Eclipse BaSyx (Sección 2.4.7 y 3.3).

---

<sup>1</sup> Llamamos *framework* a una estructura base que simplifica el desarrollo de un proyecto, ya que puede utilizarse como un punto de partida al que puede complementarse de acuerdo a lo que se desea realizar.

## 1.2 Objetivos

Este trabajo tiene una serie de objetivos donde cada uno de ellos se apoya sobre el anterior.

En primer lugar, se pretende realizar una profunda investigación para comprender en qué estado se encuentra actualmente la industria manufacturera, así como también ver hacia dónde se está dirigiendo.

En segunda instancia, detectar cuáles son los principales desafíos a los que se enfrenta actualmente el desarrollo de un sistema de CMfg.

Finalmente, realizar un aporte con respecto a la industria 4.0, desarrollando un prototipo de un sistema de CMfg que valide el uso de BaSyx para su desarrollo y mida su efectividad en entornos de fabricación en la nube.

## 1.3 Resultados

Con respecto a los resultados obtenidos, el trabajo puede dividirse en dos partes: la primera concerniente a la revisión sistemática de literatura y la segunda concerniente al desarrollo de la prueba de concepto y la experimentación asociada a ella.

Para la primera etapa de este trabajo, el principal resultado obtenido es un documento conteniendo una revisión de la literatura y del estado del arte de CMfg.

En la segunda etapa del trabajo, el resultado obtenido consiste en un sistema de fabricación en la nube que implementa una arquitectura de software basada en los resultados de la primera etapa del proyecto. Este sistema incluye también el uso de Eclipse BaSyx como *middleware* para vincular activos físicos de fabricación con sus *Digital Twins* en la nube y la demostración de algunas de las ventajas del uso de los conceptos de CMfg en ejemplos de producción realistas.

## 1.4 Estructura del documento

El resto de este documento se encuentra organizado de la forma descrita a continuación:

En la Sección 2, se presentan de forma resumida los resultados obtenidos de la revisión sistemática de la literatura acerca del estado del arte de CMfg.

En la Sección 3 se detallan los objetivos planteados para este proyecto y se realiza un abordaje más profundo de Eclipse BaSyx y sus características.

La Sección 4 desarrolla la arquitectura del sistema de CMfg diseñado para realizar pruebas de concepto, detallando en profundidad sus módulos más importantes.

La Sección 5 presenta la implementación realizada para el sistema descrito en la Sección 4.

En la Sección 6 se presenta la experimentación realizada utilizando el desarrollo de la sección anterior, incluyendo generación de casos de prueba, resultados obtenidos y análisis de los mismos.

Finalmente, se presenta la Sección 7, que contiene las conclusiones y trabajos a futuro, y el documento concluye con el Glosario, Bibliografía y Anexos.

## 2. Estado del arte

El objetivo de esta sección es mostrar de forma reducida la información obtenida durante la investigación del estado del arte de CMfg y de la Industria 4.0 en general. La versión extendida de esta investigación, que sirvió como motivación para la realización de un artículo científico que fue presentado en la revista científica *Journal of Industrial Information Integration*, así como también para la participación en distintos congresos, podrá ser encontrada en la sección de Anexos. Si bien en esta sección son abordados varios de los temas de dicho trabajo, el desarrollo de estos se verá en gran medida resumido. En caso de estar interesado en continuar leyendo material acerca del tema, se recomienda fuertemente dirigirse a la sección de Anexos y leer el material completo.

### 2.1 Contexto

Con la economía altamente globalizada y el acceso masivo a Internet, las organizaciones se están enfrentando con nuevos retos y nuevas problemáticas que no pueden satisfacerse de manera óptima con el paradigma de fabricación actual. Por otra parte, además, existe actualmente un cúmulo de características, tecnologías y conocimientos que permiten pensar en la factibilidad de reinventar la forma de fabricación utilizada por las industrias.

El concepto de IoT ha madurado en los últimos años, y está teniendo su “explosión” en la actualidad.

El Internet ya es un recurso accesible por todos y su uso se encuentra naturalizado e internalizado en la gran mayoría de las empresas. Esto, sumado al gran proceso de globalización que se ha generado, hace que las empresas puedan enfocarse en diseños de productos más complejos. Surge, también, la necesidad de satisfacer requerimientos cada vez más específicos de los consumidores. Es debido a esta necesidad que comienza a surgir el concepto de “lote de tamaño 1” [12, 65], concepto que consiste en una fabricación orientada a servicios en lugar de la típica fabricación orientada a productos.

### 2.2 Industria 4.0

Antes de continuar con el resto del documento, se hace necesario explicar el concepto de Industria 4.0. Se le llama Industria 4.0 (o Cuarta Revolución Industrial) a la cuarta etapa de las llamadas revoluciones industriales, que está teniendo lugar actualmente y consiste en la digitalización de la fabricación en toda su extensión (BaSyx / Industrie 4.0 - Eclipsepedia, [https://wiki.eclipse.org/BaSyx/\\_/Industrie\\_4.0](https://wiki.eclipse.org/BaSyx/_/Industrie_4.0)).

---

**Las revoluciones industriales a lo largo de la historia:** Previo a la actual Industria 4.0, hubo tres grandes etapas de cambios en la industria. La llamada primera revolución industrial tuvo lugar a finales del Siglo XVIII con la introducción de la mecanización del trabajo mediante el uso de máquinas a vapor. La segunda revolución industrial constó del principio de la división de trabajo y el uso de la electricidad. Finalmente, la tercera, consistió en la automatización de las tareas repetitivas, basándose en computación y robótica.

---

Entre las nuevas oportunidades que se espera alcanzar una vez finalizada esta cuarta etapa, llamada Industria 4.0, se encuentra la posibilidad de proporcionar experiencias de personalización de los pedidos para los clientes. Surge el concepto de “lotes de tamaño 1” o producción adaptativa que hace coincidir los pedidos individuales de varios clientes [12, 65], para que los pedidos individuales puedan producirse al mismo costo que los producidos en masa. Esta es una importante diferencia con respecto a sus predecesoras, ya que la cuarta revolución industrial no se centra solamente en la producción en masa, sino que se busca reducir el costo de producción por unidad para lotes pequeños.

Entre los desafíos a los que se enfrenta la Industria 4.0, la documentación oficial de Eclipse BaSyx (BaSyx / Industrie 4.0 - Eclipsepedia, [https://wiki.eclipse.org/BaSyx/\\_/Industrie\\_4.0](https://wiki.eclipse.org/BaSyx/_/Industrie_4.0)) destaca los siguientes:

1. El aumento de la eficiencia y la disminución de tiempo de inactividad para disminuir costos. Esto incluye trabajar en capacidades mejoradas de mantenimiento predictivo, utilizando datos de sensores para combinar, evaluar y utilizar para predecir los tiempos de inactividad. Pruebas virtuales de cambios en el proceso o de integración de nuevas máquinas, por ejemplo, serán también de utilidad para este desafío.
2. Mejorar la capacidad de cambio. La idea es que no se requiera un gran esfuerzo para adaptar la línea de producción a un nuevo producto y a cambios no previstos para poder finalmente trabajar de manera eficiente con tamaños de lote más pequeños.
3. Documentación automatizada. La utilización de *Digital Twins* permite seguir un producto a través del proceso de producción, recibiendo y almacenando todos los datos de calidad de forma estructurada. Esto constituye una documentación automatizada legible por una máquina y permite cumplir con los requisitos de documentación para productos personalizados. Resulta de especial interés al momento de detectar, por ejemplo, un defecto importante de seguridad después de la venta del producto. De este modo, puede ubicarse el producto en cuestión y llamarlo para mantenimiento.
4. Mejor comprensión del proceso. La estructura de una fábrica en el paradigma de Industria 4.0 permite detectar cuellos de botella en los procesos y tomar decisiones con respecto a su producción apoyándose en datos certeros.
5. Nuevos modelos de negocios. Con esta nueva revolución industrial, surgen nuevos modelos de negocios basados en datos. Se permite la creación de valor a nuevas empresas de mantenimiento predictivo y de creación de *Digital Twins*.

Sintetizando, la visión de la Industria 4.0 es la de una producción flexible, de productos personalizados en entornos de producción cambiantes con un costo similar al de los productos producidos en masa, y surge el concepto de “producción como servicio” (PaaS).

Con base en esto, surge el proyecto de desarrollar un *middleware* de código abierto para la Industria 4.0, llamado Eclipse BaSyx, para proporcionar la tecnología base necesaria para apoyar la implementación de los conceptos de Industria 4.0 en sistemas de producción.

El detalle de este *middleware* será desarrollado en la Sección 3 de este documento.

## 2.3 Beneficios de Cloud Manufacturing

Para que un nuevo paradigma sea aceptado e incorporado, tiene que ofrecer algún beneficio importante para quien lo implemente. En el caso de CMfg son las industrias quienes implementan los paradigmas de fabricación.

La presente sección pretende describir de forma general quiénes y cómo se ven beneficiados por la implementación de CMfg.

La fabricación en la nube puede ser implementada tanto por grandes empresas como por aquellas de porte pequeño y mediano, y el uso de CMfg varía en función del tamaño de cada empresa. Con respecto a esto, parece ser una oportunidad única para las pymes [16], que son las que esencialmente se beneficiarían de este paradigma, puesto que les permitiría actuar, virtualmente, como si fuesen empresas grandes. Se trata de utilizar la fabricación en la nube para implementar una fabricación cooperativa [9].

Por otra parte, al hacer uso de IoT, CMfg permite incluso interactuar desde la nube con los activos de producción. Al virtualizar los activos de las empresas, es posible monitorear el estado de los mismos, así como adelantarse a posibles fallas de funcionamiento y organizar los mantenimientos.

Esta última característica permite generar una gran base de conocimiento, obteniendo datos de una cantidad importante de empresas y dominios diferentes. De este modo, permite realizar mejores análisis, correcciones y mejoras de los procedimientos, ofreciendo la posibilidad de aprender de esta información relevada para así mejorar continuamente el sistema y optimizar los flujos.

La idea principal sobre la que se rige CMfg es la de proveer como servicios todo tipo de recursos de fabricación (XaaS<sup>2</sup>). Su principal ventaja es la vinculación automática y eficiente entre usuarios con necesidades y proveedores de recursos que puedan satisfacer dichas necesidades durante todo el proceso de fabricación.

Este enfoque permite escalar la producción muy fácilmente sin incurrir en grandes costos. Una empresa no tiene por qué comprar maquinaria nueva o más camiones o grandes almacenes, sino que puede “contratar mayores capacidades” en la nube. De la misma forma, al disminuir la producción, la empresa contrata menores capacidades en la nube y no se queda con maquinaria ociosa en su poder, lo que redundaría en gastos innecesarios. Por eso, una probable característica que debería ofrecer quien implemente un sistema de CMfg es la de “pago por uso” [7].

Finalmente, como el proceso de vinculación de proveedores y clientes es automático y cada cliente puede vincularse con un proveedor diferente para cada trabajo específico, se evitan los altos costos presentes en el paradigma de fabricación tradicional, asociado a los costos generados por los subcontratos entre empresas.

Por todo lo anteriormente mencionado, un paradigma de fabricación en la nube minimizaría muchos de los riesgos que actualmente aquejan a las empresas a lo largo de todo el ciclo de producción.

---

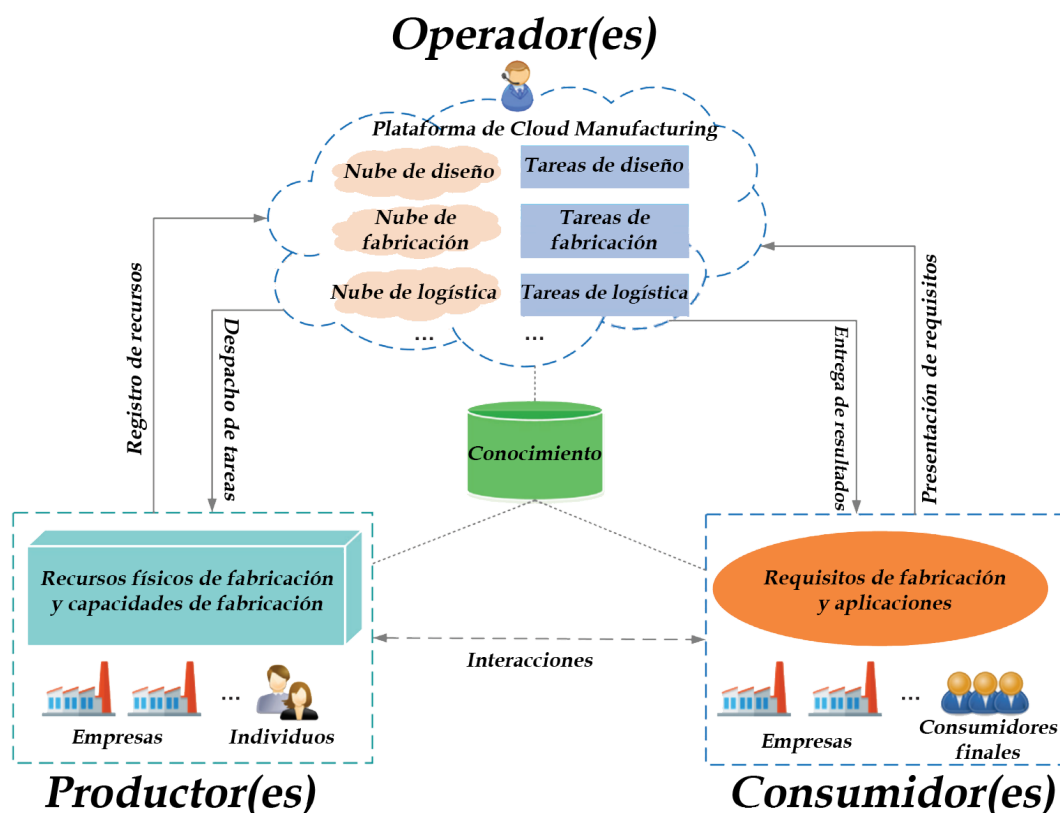
<sup>2</sup> El término XaaS proviene de “X as a Service” (o “todo como servicio”) y hace referencia a que todo recurso, servicio o producto puede ser accedido por los usuarios bajo demanda a través de Internet.

## 2.4 Características

CMfg (fabricación en la nube) es un nuevo paradigma de fabricación que incluye diseño, producción, fabricación, pruebas (testing), logística, mantenimiento y todas las demás etapas del ciclo de vida de un producto. Su objetivo es ofrecer servicios de fabricación “on-demand” a los consumidores, mediante Internet.

### 2.4.1 Actores

Según la investigación realizada, toda concepción de alguna plataforma que implemente CMfg, estará formada por cuatro grandes partes bien definidas: tres actores (o *stakeholders*) y una base de conocimiento [1]. Los clientes (o consumidores) son aquellos usuarios que necesitan obtener materia prima o servicios que no pueden -o no les rinde- producir, por ejemplo, porque la maquinaria es demasiado costosa. En esencia, tienen una necesidad que deben satisfacer, los productores (o proveedores) son aquellos usuarios que tienen los medios para satisfacer la necesidad de los clientes. Un ejemplo de esto, son aquellos que poseen maquinaria con capacidad ociosa. Los administradores de la nube, también llamados operadores, son aquellos usuarios encargados de conectar a los clientes con los productores de la manera más conveniente posible. Finalmente, se tiene la base de conocimiento, que es un tipo especial de base de datos para la gestión del conocimiento, que le permite al sistema tomar decisiones de forma eficiente. La Figura 1 ilustra esta relación entre los diferentes actores.



**Figura 1:** Modelo de operación de fabricación en la nube  
Fuente: LIU et al. (2019) [1]

## 2.4.2 Funcionalidades

En primera instancia, por ser una plataforma de fabricación, se debería permitir la conexión de una variedad de recursos de fabricación, identificando, además, información clave de los recursos conectados [9]. Más adelante, en este mismo documento, se explayará más sobre este punto, puesto que esta función sienta las bases para el resto de la operativa de la plataforma.

Relacionado con el punto anterior, una plataforma de CMfg debería también ser capaz de transformar recursos de fabricación reales en recursos virtuales. Como se desarrollará más adelante en esta investigación, una forma de virtualizar lógicamente un recurso físico es a través de los llamados “gemelos digitales” (o “*Digital Twins*”, en inglés).

Para una correcta y eficiente personalización flexible bajo demanda, estos recursos de fabricación virtualizados deberían ser agrupados para formar así conjuntos de recursos virtuales, que deberán, a su vez, estar correctamente administrados.

Una vez virtualizados y agrupados en conjuntos de recursos, la siguiente función que una plataforma de CMfg debería ofrecer, es la servitización de las capacidades de fabricación. Se le llama servitización a la transformación de las capacidades de fabricación en servicios en la nube que cumplen con una especificación estándar, añadiendo una descripción completa y efectiva de cada una de las capacidades específicas según cierta semántica [9].

Con respecto a los servicios, es preciso también contar con una forma de gestionarlos en la nube. Para ello, es necesario que el sistema provea la posibilidad de publicar servicios, realizar una vinculación (o *matching*) inteligente, componer diferentes servicios, gestionar el tiempo de ejecución y calificarlos. Análogamente a los recursos virtuales, los servicios en la nube también deberán agruparse en conjuntos (o catálogos) de servicios, que construirán la base sobre la que operarán los sistemas de fabricación en la nube

Otra funcionalidad que es también esperable de una plataforma de fabricación en la nube, consiste en el descubrimiento automático de servicios frente a las solicitudes enviadas por los clientes. Una vez que productor y consumidor confirmen una solución, la plataforma debe configurar dinámicamente un sistema de fabricación virtual mediante el ensamblaje de servicios de colaboración, así como la integración de los recursos que sean relevantes.

El emparejamiento -o *matching*- entre cliente y proveedor puede ofrecerse de dos formas diferentes, llamadas “*push*” y “*pull*”. El emparejamiento de tipo “*push*” consiste en que la plataforma de CMfg periódicamente realiza un análisis proactivo semántico y multidimensional de la información de los proveedores y de los consumidores. Luego genera sugerencias o emparejamientos inteligentes para cada uno de ellos y los resultados se envían a quienes más probablemente les serían de utilidad. En esencia, en el emparejamiento de tipo “*push*” los clientes publican sus requerimientos y esperan que el sistema les sugiera productores. En cuanto al emparejamiento de tipo “*pull*”, quienes se encargan de buscar productores en un *pool* virtual de productores son los clientes.

En este sentido, es necesario también garantizar que el sistema de fabricación virtual se ejecute según lo esperado. Para ello, la plataforma en la nube debe monitorear el estado de



tiempo de ejecución y responder a los cambios. Según los procesos de tareas de fabricación, será necesario coordinar diferentes proveedores de servicios entre los servicios, ya sean servicios *OnCloud* u *OffCloud*. Por otro lado, también debe proporcionar funciones de autodiagnóstico y autoreparación, además de autoadaptación ante excepciones que ocurran en tiempo real.

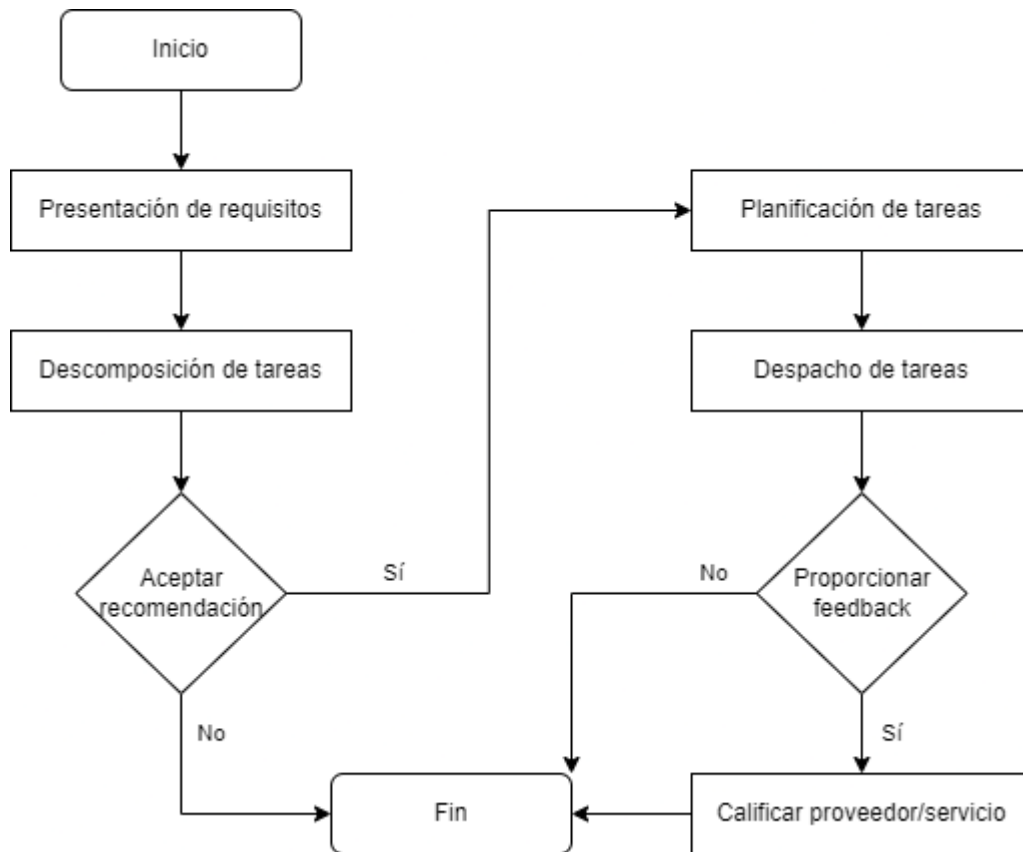
### 2.4.3 Tareas que debe desarrollar un sistema CMFg

La revisión de literatura arroja una serie de tareas claves que un sistema de CMfg debería proveer, que podrían resumirse en las once siguientes [1]:

1. **Registro de recursos (Resource registration):** los fabricantes deben tener la posibilidad de ofrecer sus servicios/recursos a través de la plataforma.
2. **Virtualización de recursos (Resource virtualization):** virtualizar los recursos que son ofrecidos por los proveedores.
3. **Presentación de requisitos (Requirement submission):** los consumidores de recursos necesitan poder especificar en la plataforma de CMfg todos los requisitos para llevar a cabo su producción.
4. **Descomposición de tareas (Task decomposition):** descomponer en subtareas con complejidad inferior aquellas tareas más complejas que no puedan ser desarrolladas por un único servicio.
5. **Planificación (Scheduling):** proceso de organización, control y optimización del trabajo que resulta ser una tarea de vital importancia para CMfg. Es el proceso de asignación de recursos/servicios a tareas, el monitoreo, control y optimización de estados de servicios y recursos y la ejecución de tareas para satisfacer los requisitos individualizados de los consumidores. Incluye el procesamiento de tareas -como la descomposición de ellas, por ejemplo-, el descubrimiento de servicios, y demás. Se profundizará sobre el proceso de planificación en la Sección 5.1 de este documento y en el Anexo I (Estado del arte).
6. **Despacho de tareas (Task dispatching):** capacidad de asignar un trabajo a un recurso. Una vez realizada la descomposición de tareas, cada sub tarea debería ser asignada a un servicio, según lo recomendado en la etapa de planificación.
7. **Entrega de resultados (Result delivery):** proveer al usuario y al sistema el resultado de la ejecución del trabajo una vez que este finaliza.
8. **Recomendación y descubrimiento de clientes (Customer recommendation and discovery)**
9. **Recomendación y descubrimiento de proveedores/servicios (Provider recommendation and discovery/Service recommendation)**
10. **Generación y mantenimiento de una base de conocimiento:** el *scheduler* hace un fuerte uso del conocimiento adquirido previamente. Mientras mayor sea este, mejores serán los resultados obtenidos por dicho *scheduler*, es por ello que es importante contar con una base de conocimiento grande.
11. **Herramientas para manejar y operar el sistema:** proveer distintas interfaces para que tanto clientes y fabricantes como operadores puedan interactuar con el sistema.

Los diagramas de Figura 2 y Figura 3 muestran dos de los principales casos de uso que el sistema debería proveer a productores y consumidores.

La primera figura corresponde al flujo típico de un caso de uso de un consumidor de recursos de producción. El usuario deberá ser capaz de publicar sus requisitos de producción, que deberán satisfacerse mediante servicios en la nube.



**Figura 2:** Diagrama de flujo de tareas de un sistema de CMfg que forma parte de un caso de uso típico para el consumo de servicios por parte de un usuario de tipo cliente. Fuente: elaboración propia.

La segunda figura, por otra parte, muestra un caso de uso típico de los productores (usuarios que ofrecen capacidad de fabricación), donde se registra y virtualiza recursos que los proveedores ofrecen para ser utilizados por los clientes.



**Figura 3:** Diagrama de flujo de tareas de un sistema de CMfg que forma parte de un caso de uso típico para el registro y virtualización de recursos por parte de un usuario de tipo proveedor.  
Fuente: elaboración propia.

#### 2.4.4 Ejemplos de uso

De acuerdo a la literatura, existen una gran variedad de entornos donde CMfg puede ajustarse muy bien y realizar una mejora significativa en la productividad. A continuación se mencionan algunos de casos de uso que se pudieron encontrar durante la revisión de la literatura.

Según Simeone et al. [28], una industria que muy probablemente se beneficiaría de esto, es la industria de corte de chapa (sheet metal cutting industry). En este ambiente los cambios frecuentes de volúmenes de demanda y variedad de productos, afectan negativamente la eficiencia de los recursos. Reducir el desperdicio de material es posible aplicando una estrategia de anidación (combinar múltiples órdenes de corte de chapa en la misma chapa). CMfg puede ayudar a realizar esto, facilitando la configuración de redes de fabricación conectando clientes y proveedores a lo largo del globo mediante una plataforma común.

Simeone et al. [26] expande el ejemplo de arriba, analizando esta vez desde el punto de vista de los productores. La estrategia para el mejor aprovechamiento de materiales por parte de los productores (minimizar el desperdicio de materiales) se realiza mediante un Genetic Algorithm. La idea es organizar los pedidos de los consumidores de forma tal de utilizar el mayor porcentaje de la placa de chapa. La página 4 del pdf del mencionado artículo explica, a alto nivel, cómo funcionaría el motor de apoyo a la toma de decisiones. En dicho artículo, se realizó una simulación utilizando dos productores y dos consumidores, mostrando las diferentes formas de organizar el trabajo (incorporar, encolar, priorizar, rechazar consumidores). Este artículo tuvo en cuenta, además de la maximización del porcentaje de utilización de las materias primas, las deadlines de los consumidores.

Adicionalmente, se mostró que el proceso de optimización es costoso en materia de recursos y tiempo de ejecución.

#### 2.4.5 Temas a resolver

Como CMfg y el concepto de Industria 4.0 es relativamente nuevo en el tiempo, existen ciertos temas para los que aún no se ha encontrado consenso. Al momento de nombrar los diferentes temas principales que se deben resolver, el primero que se hace necesario subrayar es el de definir exactamente qué es CMfg. No existe aún una definición internacional precisa, aceptada tanto por la academia como por la industria [2, 7, 29]. Adicionalmente, del proceso de investigación se desprende también que no existe aún una arquitectura, plataforma, modelo, framework o aplicación que implemente CMfg ni se ha estandarizado su diseño. Existen, no obstante, algunas soluciones que diferentes investigadores han planteado, como VEO<sup>3</sup> [12, 13, 14] y *Digital Twins* [4, 30, 31, 32, 33, 35, 36].

Una vez definido el concepto y la arquitectura, surgen temas subyacentes que también se encuentran en proceso de investigación, discusión y resolución.

La utilización de nubes públicas o comunitarias para fabricación cooperativa plantea la interrogante acerca de quién es realmente el dueño de la información -posiblemente confidencial- de la parte virtual de los activos de las empresas.

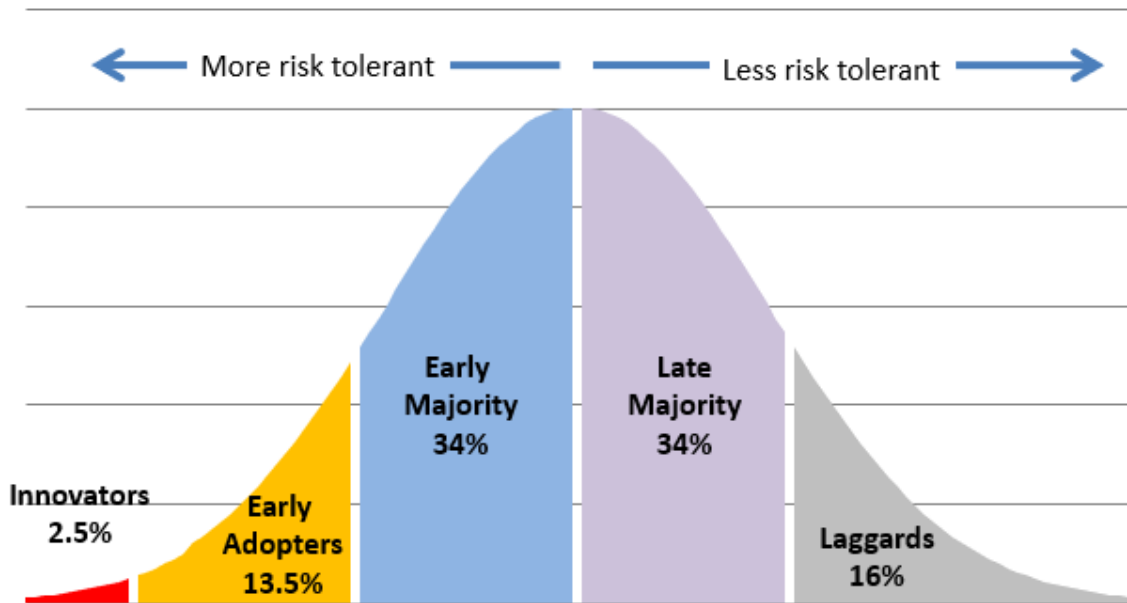
La estandarización de hardware y software para integración entre sistemas es también un tema fundamental a resolver con el fin de disminuir costos de implementación y maximizar eficiencia y seguridad. Esto incluye, pero no se limita, a protocolos de I/O y estructuras para comunicación horizontal en lugar de vertical.

Adicionalmente, como cualquier proceso de adopción de nuevas tecnologías y paradigmas, es preciso resolver la adopción de CMfg y cómo poblar la nube. Por los beneficios que CMfg ofrece, se estima que, una vez que el sistema se encuentre funcionando, las empresas querrán utilizar el paradigma de Industrias 4.0. Las primeras empresas -*Innovators* y *Early Adopters*<sup>4</sup>-, sin embargo, deberán ser incentivadas, ya que se estima que tendrán preocupaciones concernientes a seguridad, costo y estrategia de adopción (Figura 4) [2].

---

<sup>3</sup> VEO: Virtual Engineering Objects.

<sup>4</sup> Fuente: "Basics of The Adoption Curve - Fundamentals of B2B Product Management Series: Part 3", Michael Mullany (<https://www.linkedin.com/pulse/basics-adoption-curve-michael-mullany>)



**Figura 4:** Porcentaje de clientes que adoptan una innovación a lo largo del tiempo  
Fuente: Mullany (2018) [48]

En este sentido, la adopción de CMfg requerirá un nuevo modelo de negocio que apoye la naturaleza colaborativa de este paradigma, donde socios en red cooperan en la distribución de valor.

Finalmente, la alta automatización en las industrias 4.0, el uso de robots, IA, aprendizaje automático y demás, hacen que se deba reformular qué roles cumplirían los seres humanos.

#### 2.4.6 Casos de estudio más frecuentes

##### Virtualización

Una de las bases de CMfg consiste en poder conectar el mundo físico con el virtual, por lo que resulta de especial interés contar con una herramienta que permita virtualizar recursos de producción.

Los Cyber Physical Systems (CPS), se definen como tecnologías transformadoras cuyo fin es administrar sistemas interconectados entre sus activos físicos y sus capacidades computacionales [34]. En las fábricas inteligentes, la visibilidad y trazabilidad en tiempo real tienen un rol fundamental. Los recursos de fabricación típicos se convierten en objetos de fabricación inteligentes (SMO) mediante el uso de dispositivos IoT como lectores y etiquetas RFID [34]. Existen varias propuestas que van en esta dirección en la literatura analizada [34], donde destaca el concepto de *Digital Twin* (DT). En el Anexo I se realiza un abordaje en mayor profundidad sobre el concepto de *Digital Twin*.

La tendencia de cantidad de publicaciones acerca de virtualización de recursos de producción es creciente, indicando que es un tema importante a abordar. Se espera que en

el corto plazo sean propuestas más arquitecturas de software y aplicaciones que permitan virtualizar recursos.

### **Compartición de recursos**

Además de virtualizar los recursos de producción, un tema importante muy tratado en la literatura es el hecho de compartir recursos (resource sharing).

Compartir los recursos de producción impacta positivamente en la eficiencia de la organización. Reduciendo el tiempo que el recurso permanece en desuso, buscando minimizarlo (tratar de reducir al máximo posible la capacidad ociosa).

No se desprende de la investigación que exista actualmente un consenso sobre la forma en la que los recursos de producción puedan ser compartidos eficientemente en ambientes de CMfg, sin embargo, existe una gran cantidad de artículos a este respecto.

Por ejemplo, [37] propone utilizar una plataforma inteligente de fabricación en la nube para permitir el intercambio dinámico e inteligente de, en este caso, los servicios de corte de chapa en una red de fabricación. La solución propuesta, cuyo detalle puede leerse en [37], ofrece soluciones de fabricación inteligente bajo demanda. Gestionando y combinando de manera óptima y dinámica, varias instancias de los servicios que son ingresadas (conectadas globalmente a la plataforma común) por clientes y proveedores. Esta plataforma habilitaría la compartición de recursos físicos entre diferentes organizaciones. Sin embargo, es posible también compartir recursos -tanto físicos como no físicos- dentro de una misma empresa, como se indica en [38]. En este se detalla un sistema de similares características al anterior y se describe una arquitectura estructurada en tres niveles. Esta arquitectura está destinada a ofrecer servicios on-demand para evaluar la calidad de productos, ejemplificándose con el caso de uso de la industria aeronáutica.

### **Planificación**

Como se ha ido mostrando a lo largo de este documento, en un sistema de CMfg existen múltiples recursos. Cada uno de ellos pueden llegar a ser utilizados por múltiples dispositivos. También existen muchos dispositivos que brindan un mismo servicio. Es por ello que es de vital importancia poder administrar todos estos elementos para que puedan trabajar en conjunto y de forma eficiente sin generar conflictos entre ellos. Es de ahí que surge otro de los problemas más abordados por la literatura, que consiste en la planificación tanto de recursos, como de servicios.

De la investigación llevada a cabo surgieron diferentes trabajos que atacan esta problemática.

Por ejemplo, en [39], se propone una arquitectura de Multi-agent la cual consiste en un sistema de planificación multi agente a nivel de plataforma y otro a nivel de empresa. Por otro lado [43], presenta una arquitectura "semi heterarchical". Esta permite la integración de diferentes niveles de gestión. Cada uno de ellos con su cuota de autonomía en la toma de decisiones y responsabilidades específicas (negocio, desempeño general, operativo).

Así como estos, existen otros trabajos, donde cada uno de ellos busca presentar su propia versión y solución a este problema, como por ejemplo [42, 41, 40], cuyo desarrollo se encuentra en la sección de Anexos.

### **Almacenamiento de datos**

Aprovechando los grandes volúmenes de información que son recopilados con este paradigma, y que sirven como entrada para procesos de aprendizaje automático y mejora continua, surgen ciertas consideraciones. Una de las consideraciones se refiere a los grandes costos que conlleva el almacenamiento y mantenimiento de estos datos [44].

Un caso de uso estudiado [44], propone una arquitectura jerárquica de tres niveles para almacenamiento de series temporales de datos, que ayuda a reducir costos. El primer nivel consta de dispositivos no volátiles (como SSD) para almacenar por un corto período las nuevas series de datos sin procesar. El segundo nivel, consta de discos duros magnéticos (HDD) para almacenar durante un período medio las series de datos de tiempo recientes. Finalmente, un tercer nivel almacenaría una representación reducida de las series de tiempo, utilizando técnicas de reducción y almacenándolas en discos duros por períodos más prolongados.

Con respecto al almacenamiento de datos también surge el problema de posibles filtraciones de datos a lo largo de estas etapas. Existe la posibilidad de cifrar datos sensibles que serán almacenados en la nube, pero se eliminaría la posibilidad de realizar búsquedas de manera eficiente [45]. El paper referenciado, propone una variedad de técnicas con el fin de acelerar el rendimiento de búsqueda de datos cifrados en la nube.

### **Supervisión**

Resulta también de especial interés el hecho de poder llevar el control de la producción y de la maquinaria en todo momento. El monitoreo de los sistemas de producción y del ciclo de vida de los productos permite tomar decisiones sobre la marcha, ajustar procedimientos, predecir mantenimientos y evitar fallas, reduciendo riesgos y disminuyendo costos. El proceso de supervisión se complementa con el de análisis de datos (siguiente caso de uso). Los procesos de supervisión actuales suelen ser manuales y poco precisos, o quedan rezagados en cuanto a recopilación de datos. BA [46] propone un modelo de recopilación de datos en vivo para producción, rechazo y tiempo de inactividad, apoyándose en sensores electrónicos y en IoT. Canizo [47], por otra parte, propone una plataforma a gran escala para el monitoreo de CPS en tiempo real, basado en Big Data.

Como se remarca en toda la extensión de este documento, la supervisión (*monitoring*) es una actividad crítica en CMfg y es abordada en múltiples artículos [17, 18, 19, 20, 21, 27], siendo su diseño y desarrollo un desafío en sí mismo.

### **Análisis de datos**

Existe una gran importancia en la realización de un correcto análisis de datos para que el desarrollo de CMfg pueda realizarse exitosamente en la industria. Los grandes volúmenes de datos recopilados no sirven de nada si no se aprovechan como “materia prima” para ser analizados.

No resulta extraño, en este sentido, encontrar una gran cantidad de trabajos de investigación cuyo eje temático es, justamente, el análisis de datos.

Ya sea para mantenimiento predictivo [22, 25, 15], control y optimización de costos de producción [27] y de logística [11], detección y corrección de fallas [18, 19, 25], optimización



de procesos [21], predicción de tiempo de vida útil [15], reconfiguración y adaptación mediante re-parametrización óptima [23] y otras acciones.

Los trabajos citados en el párrafo anterior, proponen distintas arquitecturas de software y soluciones para no solo recopilar, sino también analizar big data en entornos de fabricación en la nube. Técnicas de aprendizaje automático [24] son fuertemente utilizadas en este respecto, así como otras técnicas nuevas, como el DEDG<sup>5</sup>, apoyado en técnicas de deep learning [15].

### **Seguridad y optimizaciones**

Como fue mencionado en 2.4.5, una de las grandes preocupaciones de los sistemas en la nube es el tema de la ciberseguridad y la privacidad. Conceptos como la transparencia de datos que juegan un factor muy importante en la industria 4.0 resultan en consecuencias potencialmente importantes para los ciberataques exitosos [8].

Es por ello que algunos investigadores no han escatimado esfuerzos a la hora de tratar de resolver esta problemática. [8] propone un esquema de seguridad que modela las interacciones dinámicas de los componentes de la industria 4.0. Por otro lado, [10] presenta un framework de ciberseguridad y resiliencia para aplicaciones de fabricación basadas en redes definidas por software.

En un sistema de CMfg, como ya fue mencionado previamente, se manejan grandes volúmenes de datos y procesamientos. Por ello es importante mantener el sistema lo más optimizado posible de forma tal de reducir las demoras lo máximo que sea posible.

Para ello, [5] propone una nueva arquitectura para el procesamiento de grandes cantidades de datos llamada Phi architecture. Esta arquitectura está diseñada para un patrón de microservicios, mejorando de esta forma la flexibilidad y estabilidad de la arquitectura.

[6] presenta una solución distinta, la cual consiste en un enfoque sistemático y comprensivo, basado en una arquitectura de tres niveles. Teniendo como objetivo, el comprender modelos predictivos de procesos de fabricación a partir de fuentes dispares de información, analíticas, y experimentales.

El manejo de materiales es una parte esencial de una fábrica inteligente, por ello [3] presenta y analiza una entidad industrial cognitiva llamada context aware cloud robotics (CACR). Esta se caracteriza por sus servicios sensibles al contexto y por su balanceo de carga efectivo.

---

<sup>5</sup> DEDG: sigla utilizada por Yan et al. para referirse al concepto de “Device Electrocardiogram”

## 2.4.7 Plataformas para Industrias 4.0

De la investigación realizada se desprende la existencia de una plataforma que permite la implementación de varios conceptos de Industrias 4.0 mediante la utilización de algunos SDKs<sup>6</sup> para diferentes lenguajes de programación.

Realizar una investigación acerca de esta nueva plataforma resulta, entonces, de especial interés, por lo que en esta sección se abordará el estudio de los conceptos de BaSys 4 y su implementación de referencia, Eclipse BaSyx.

### **BaSys 4**

BaSys 4 consiste en un middleware que define una arquitectura de referencia para sistemas de producción que permite el cambio a la industria 4.0.

El proyecto de BaSys fue financiado con 12 millones de euros por el Ministerio Federal de Educación e Investigación de Alemania (BMBF<sup>7</sup>) y se extendió durante tres años, desde el 1° de julio de 2016 hasta el año 2019.

Una vez calificada como exitosa la primera etapa del proyecto, el BMBF financió con 11.3 millones de euros la segunda parte del mismo, BaSys 4.2, durante los siguientes tres años. A la fecha de realización de este documento, se continúa trabajando sobre este proyecto.

### **Eclipse BaSyx**

Eclipse BaSyx es una implementación de referencia de los conceptos de BaSys 4.0 y BaSys 4.2 en una plataforma de código abierto desarrollada por BaSys en conjunto con la Fundación Eclipse.

BaSyx cuenta con un kit de desarrollo de software de código abierto (SDK) mediante el cual permite a los desarrolladores implementar gran parte de las características de CMfg relacionadas a la virtualización de dispositivos físicos.

A continuación, se explicará alguna de las funcionalidades que provee BaSyx para la implementación de los *Digital Twins*.

### **Modelado de Objetos**

Una de las principales ventajas de BaSyx es el conjunto de herramientas que provee para representar elementos claves de la industria 4.0. Entre este conjunto se destacan fundamentalmente, dos clases: Asset y Asset Administration Shell (AAS).

La clase Asset permite modelar objetos de la realidad (activos). Gracias a ella es posible crear *Digital Twins* para encapsular toda la información relevante asociada al activo que se quiere representar, así como también los distintos servicios que ofrece dicho activo.

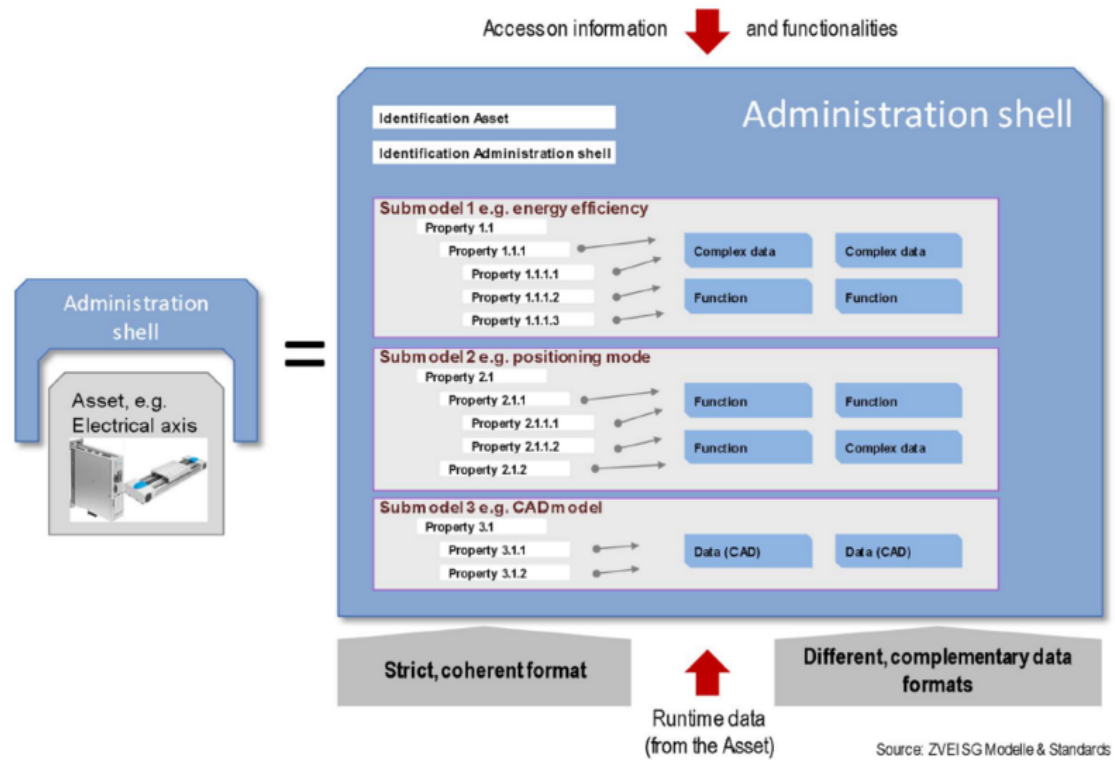
Los Assets son encapsulados en Asset Administration Shells (AAS). Como muestra la Figura 5, estos proveen la interfaz para hacer uso de los distintos servicios asociados a

---

<sup>6</sup> SDK: Kit de desarrollo de software (Software development kit)

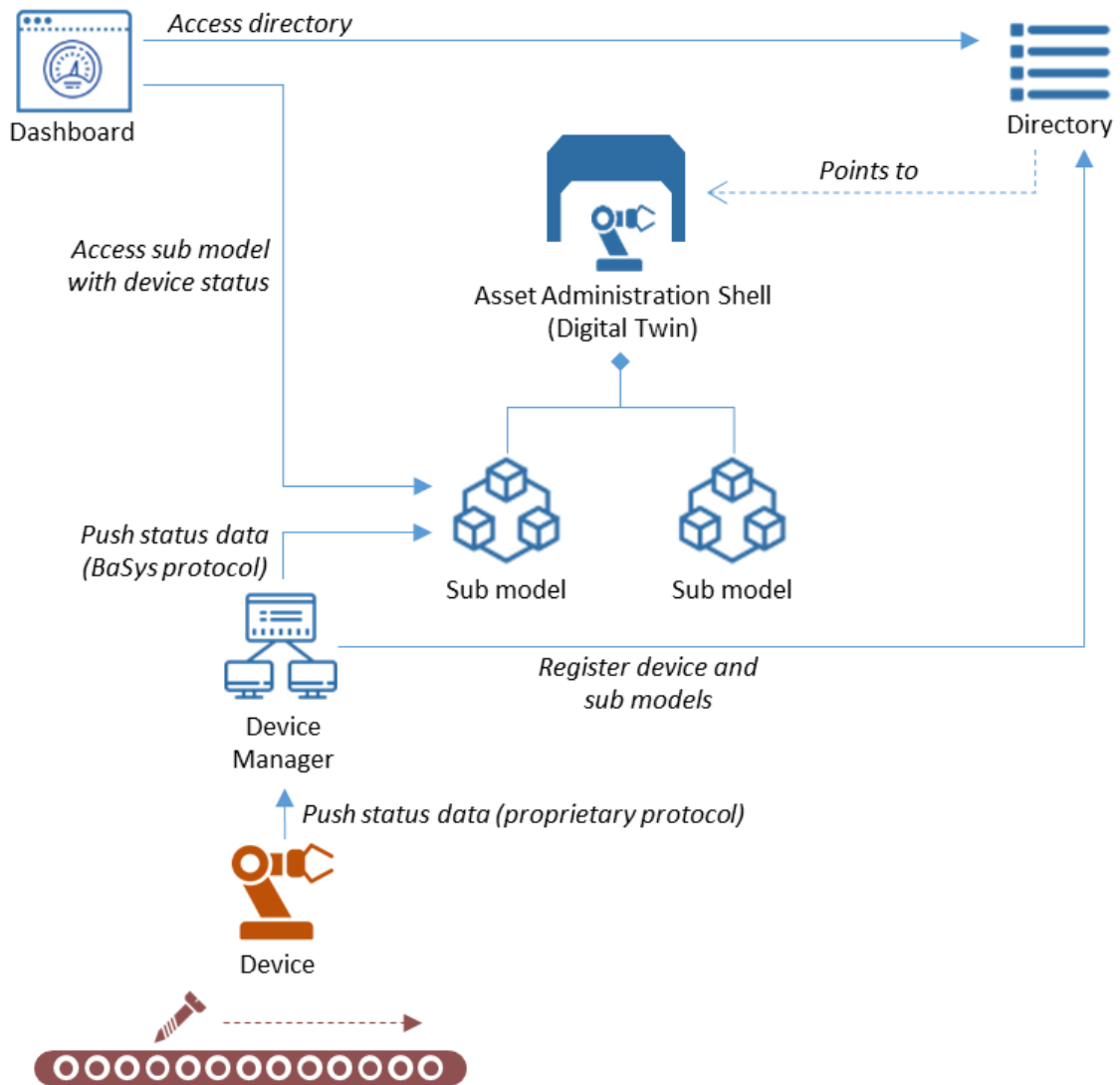
<sup>7</sup> Sigla en alemán: *Bundesministerium für Bildung und Forschung* (Fuente: <https://www.bmbf.de/>)

dicho asset. Para ello, la clase AAS ofrece la posibilidad de crear distintos submódulos donde publicar distintos servicios provistos por el Asset. Esto permite realizar una interacción con el objeto representado a través de un conjunto organizado de servicios y propiedades.



**Figura 5:** Composición de un Asset Administration Shell en BaSyx  
Fuente: Kuhn (2019) [49]

La Figura 6 a continuación muestra, de forma simplificada, una arquitectura de BaSyx 4.0. En ella se hace uso de un AAS con un par de submodelos. Además, ofrece un directorio que es utilizado por un tablero (dashboard) para poder acceder a los diferentes Asset Administration Shells.

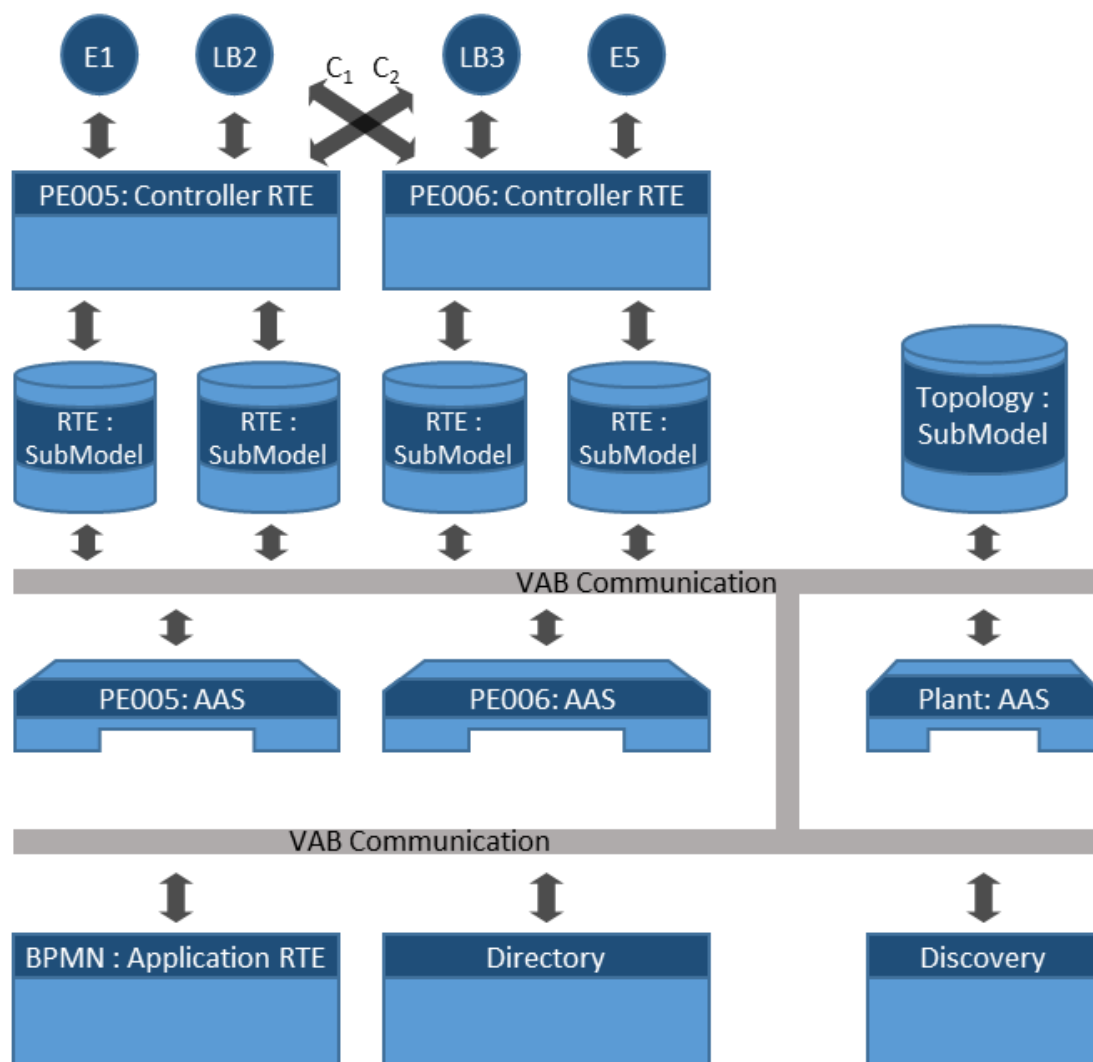


**Figura 6:** Ejemplo de arquitectura de Asset Administration Shell en BaSyx  
Fuente: Kuhn (2019) [50]

Hoy en día existen una gran variedad de dispositivos tecnológicos que implementan herramientas de IoT, donde cada uno de ellos utiliza protocolos para conectarse a la red. Dada la gran variedad de protocolos y dispositivos que existen hoy en día, surge la necesidad de poder comunicarse con distintos protocolos. Para ello BaSyx provee una herramienta llamada Virtual Automation Bus (VAB), cuyo fin es proveer una comunicación end to end para la industria 4.0.

Esta implementación permite mapear una semántica de comunicación con cinco primitivas (create, delete, retrieve, update, invoke) a diferentes redes y protocolos.

Cada protocolo puede asignarse al VAB mediante la creación de una asignación de estas cinco primitivas a sus respectivas técnicas de comunicación nativas. Con esto, se reduce el esfuerzo necesario para adaptar distintos tipos de tecnologías de comunicación.



**Figura 7:** Concepto de Virtual Automation Bus (VAB) con entidades conectadas  
Fuente: Kuhn (2018) [51]

La Figura 7 muestra un diagrama de concepto de VAB con algunas entidades conectadas, donde:

- Los *Controller runtime environments (Controller RTE)* son entornos de ejecución en tiempo real que están especialmente diseñados para el control automático. En este ejemplo, tanto PE005 como PE006 representan controladores lógicos programables (PLC)<sup>8</sup> que son un tipo de *Controller RTE*, a su vez ambos cuentan con sus respectivos AAS, llamados PE005:AAS y PE006:AAS. Estos implementan, por ejemplo, controladores de dispositivos que proveen acceso basado en servicios a dispositivos controlados. En el ejemplo mostrado en la Figura 7 estos serían motores actuadores (E1 y E5), así como barreras livianas (LB2 y LB3)
- *Application runtime environments (Application RTE)*, ejecuta aplicaciones de alto nivel las cuales interactúan con la planta. A su vez también implementan componentes de grupos, los cuales realizan servicios de alto nivel basados en servicios de otro componente de grupo o dispositivo. Las aplicaciones RTE contienen aplicaciones que ofrecen y utilizan llamadas de servicio para interactuar con otras funciones. Requieren acceso al VAB para comunicarse con los dispositivos de la planta.
- El *discovery service* es una forma unificada para que los componentes se conecten con entidades VAB y descubran AAS.
- Los elementos VAB proporcionan una interfaz unificada para acceder a las entidades que están conectadas al VAB. Admiten el acceso de lectura / escritura a los valores e implementan la operación invocable, así como la creación dinámica opcional y la eliminación de propiedades.

---

<sup>8</sup> Un Controlador Lógico Programable (o PLC, por su sigla en inglés) es una computadora utilizada en la automatización industrial para abordar los problemas de un proceso de ensamblaje en particular.

### 3. Eclipse BaSyx

Según la documentación oficial de BaSyx, que puede encontrarse en (BaSyx - Eclipsepedia, <https://wiki.eclipse.org/BaSyx>), BaSyx es una plataforma de código abierto que facilita la implementación de la industria 4.0 para pequeñas y grandes empresas. BaSyx aborda los cambios de los procesos de producción como uno de sus objetivos principales.

La producción cambiante de Industria 4.0 incluye los cambios no planificados de los procesos de producción, como por ejemplo, variantes de productos que no se conocían cuando se creó y comenzó la línea de producción.

Por otro lado, la producción flexible hace referencia a una línea de producción con un rango predefinido de flexibilidad, como por ejemplo, una cantidad bien conocida de variantes de productos. Cambiar una producción, requiere la intervención (manual) de la línea de producción. El principal objetivo de BaSyx es reducir al mínimo el tiempo de inactividad resultante.

Una producción cambiante habilita la fabricación de diferentes productos o recetas bajo la misma línea de producción, donde una receta es un plan que describe los pasos de producción que son necesarios para la fabricación de un producto, así como agregar y remover dinámicamente productos, recursos y capacidades de dispositivos.

La plataforma de Eclipse BaSyx provee componentes comunes y reusables de la industria 4.0 así como también SDKs que soportan el desarrollo de nuevos componentes para la Industria 4.0, y de esta forma permitir un rápido desarrollo de soluciones para sistemas de CMfg.

Para ello, el *middleware* de BaSyx implementa los conceptos principales de la Industria 4.0 y provee los componentes de software necesarios para implementar varias aplicaciones. Entre los conceptos implementados, se encuentran:

- Digitalización de extremo a extremo de la producción
- Conectividad de extremo a extremo entre el taller e IT
- Comunicación de igual a igual entre los dispositivos y sus *Digital Twins*
- Seguimiento de los *Digital Twins*, flujo de datos y productos
- Procesos de producción cambiantes hasta la producción de tamaño de lote 1
- *Digital Twins* para procesos, productos, dispositivos y más
- Análisis de *big data* de los procesos de producción
- Contenedores preconfigurados para aplicaciones de Industria 4.0, por ejemplo, mantenimiento predictivo
- Trazado y documentación automática de procesos de producción

### 3.1 Por qué usar Eclipse BaSyx en un sistema de CMfg

Eclipse BaSyx, como middleware que soporta la implementación de ambientes de producción bajo la industria 4.0, conecta activos de producción relevantes (*assets*) e implementa todos los componentes necesarios para la digitalización de extremo a extremo de CMfg. También soporta, implementa y constantemente evalúa estándares existentes o en desarrollo, por lo que no hay necesidad de re-implementar dichos estándares.

Esta plataforma integra numerosas herramientas de código abierto que crean un ecosistema de Industria 4.0 y entrega aplicaciones prontas para ser utilizadas.

Los conceptos principales de BaSyx son los *Asset Administration Shells* (AAS), *AAS Sub Models*, y componentes de control. Los AAS y *AAS Sub Models* son desplegados en un servidor de AAS, y pueden ser localizados mediante directorios, permitiendo la persistencia al utilizar almacenamiento en el backend. Los datos existentes pueden integrarse como *AAS Sub Models* para proporcionarlos a través de una interfaz unificada.

Los componentes de control implementan una interfaz unificada y basada en servicios para los dispositivos, los *AAS Sub Models* describen los servicios ofrecidos por los dispositivos disponibles. La interfaz basada en servicios permite la producción a medida de cada producto, donde cada producto y pieza de trabajo puede crearse utilizando su propia receta única. Los tamaños de lote pequeños son posibles, ya que cada pieza de trabajo puede definir un conjunto único de parámetros y pasos de producción.

El VAB permite la comunicación de un extremo a otro entre todos los dispositivos y sistemas de IT.

Todos estos componentes mencionados anteriormente pueden ser utilizados de forma independiente entre sí, y es gracias a estos componentes que BaSyx provee un amplio soporte y facilidad para el desarrollo de sistemas de CMfg.



## 3.2 Componentes principales

En esta sección se definirán los componentes e interfaces principales, así como también se especificará cómo interactúan entre ellas.

Para ello es importante aclarar que BaSyx consta de 4 capas bien definidas: el nivel de campo (**field level**), el nivel de dispositivo (**device level**), el nivel de *middleware* (**middleware level**) y el nivel de planta (**plant level**).

La primera capa, **field level**, consta de dispositivos de automatización, sensores y actuadores sin una interfaz específica de BaSys.

En la capa **device level** se encuentran los dispositivos de automatización que ofrecen una interfaz con base en BaSys 4.0. Aquí también se encuentran los dispositivos de puente que implementan interfaces compatibles con BaSys 4.0 para dispositivos que no proporcionan una interfaz compatible por sí mismos.

Luego, en la capa **middleware**, se sitúan los componentes reusables de la Industria 4.0 que implementan las capacidades genéricas e independientes de la planta para la línea de producción de la industria 4.0.

Finalmente, en la capa de **plant level** se encuentran los componentes de alto nivel de la planta, encargados de manejar, optimizar y monitorear la producción

En la Figura 8 se pueden apreciar los distintos componentes de BaSyx, ubicados en las distintas capas y cómo estos interactúan entre ellos.

Los dispositivos y los componentes de control asociados se encuentran en la capa **field level** y **device level** respectivamente. A su vez, los dispositivos y, por lo tanto, los componentes de control poseen un AAS asociado en la capa **middleware**.

Los AAS como fue mencionado, contienen submodelos que proporcionan información, incluyendo servicios ofrecidos y valores de estado en tiempo real. Los componentes de control registran sus servicios ofrecidos en submodelos AAS y la información en tiempo real sobre el estado de los componentes o los valores del sensor también se encuentran disponibles a través de los submodelos AAS

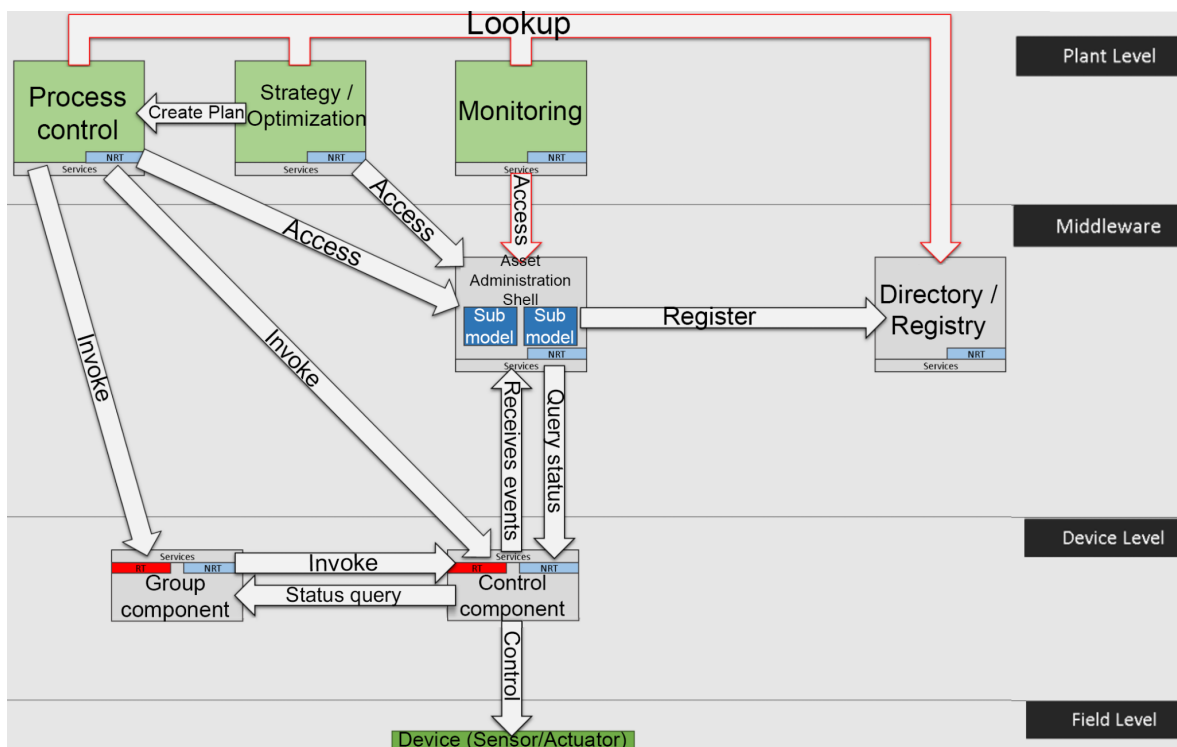
Luego los AAS son registrados en un directorio, de forma tal que las aplicaciones dependientes u otros AAS puedan ubicarlos.

Los **group components** de la capa **device level** son componentes de control que no interactúan directamente con los dispositivos, en cambio, interactúan con otros componentes de control y de grupo para proporcionar servicios de nivel superior. Los componentes del grupo también tienen un AAS asociado.

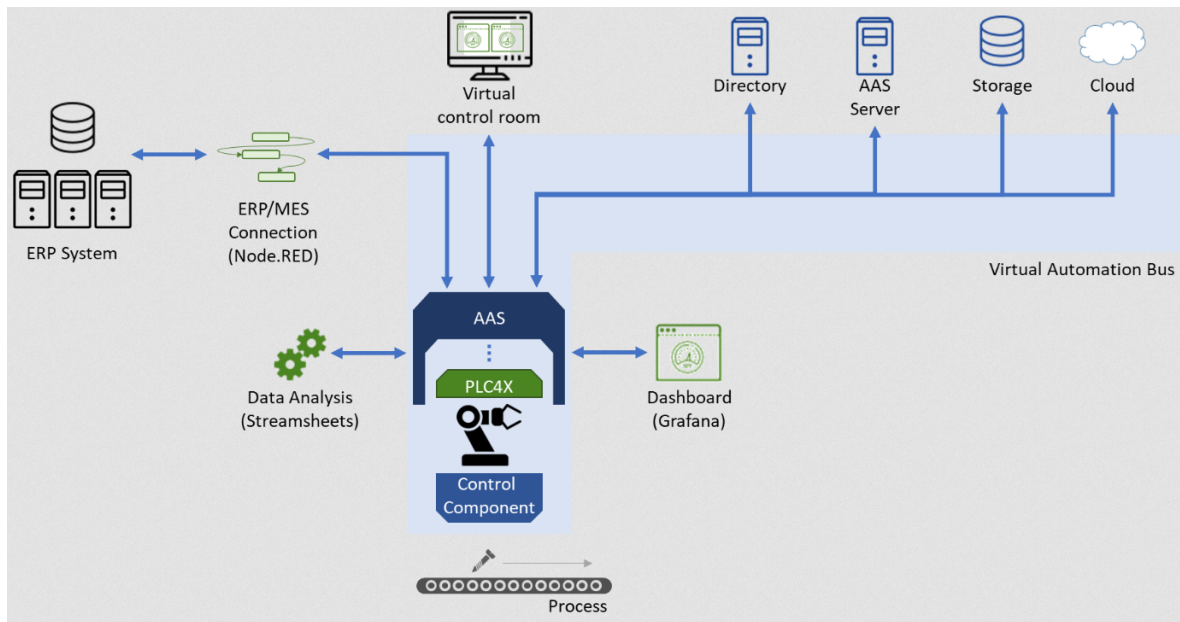
El componente *Strategy* ubicado en la capa de **plant level** es el encargado de crear un plan de producción para un producto. Para ello, busca en el directorio el producto y el AAS asociado al dispositivo de su producción. A través del AAS recibe toda la información necesaria para vincular los pasos de producción del producto solicitados con los dispositivos de producción desde el dispositivo AAS. Finalmente, haciendo uso de toda esta información, el *Strategy* crea un plan de producción para el producto dado, plan que contiene información de cuándo y qué dispositivo realizará cada paso de producción.

Una vez elaborado el plan de producción, este es enviado al componente de **process control**, que también se encuentra en la capa **plant level**. Este componente es el encargado de ejecutar el plan de producción definido por *Strategy*.

Y ya como último componente a mencionar, se presenta el componente **monitoring**, dicho componente ubicado en la misma capa que los dos mencionados anteriormente, provee la posibilidad de monitorear los distintos AAS, obteniendo sus datos, analizándolos y condensándolos para facilitar su evaluación en tiempo real.



**Figura 8:** Imagen que describe cómo interactúan los componentes de BaSyx  
Fuente: Kuhn (2018) [52]



**Figura 9:** Principales componentes de BaSyx  
Fuente: Kuhn (2018) [53]

### 3.3 Utilizando BaSyx en un sistema de CMfg

Con el fin de evaluar el desempeño y las ventajas del uso de BaSyx para la implementación de un sistema de CMfg se decidió realizar un prototipo que permita simular un entorno de CMfg.

Para ello se decidió implementar una aplicación web que permita tanto la creación de nuevas máquinas y órdenes, como la monitorización de su estado y progreso.

Esta aplicación web consume una API que hace uso de las primitivas y componentes que BaSyx provee. Adicionalmente, la API implementa dos módulos que simulan los componentes de alto nivel de la planta: el módulo de planificación y el módulo de verificación.

El primer módulo es el llamado *scheduler*, cuyo fin es evaluar las nuevas órdenes de trabajo recibidas por la aplicación, descomponerlas en tareas más pequeñas y distribuir las entre las distintas máquinas del sistema con base en el algoritmo de planificación desarrollado. Para realizar esto, es importante conocer el estado actual de las máquinas así como también poder comunicarse con ellas. Estas necesidades pueden ser fácilmente resueltas con BaSyx, aprovechando la comunicación punto a punto provista por el VAB, que permite hacer uso de las primitivas de HTTP para obtener y enviar datos entre los dispositivos de producción y el sistema.

El segundo módulo implementado consiste en el módulo *monitoring*, que es el encargado de estar continuamente monitoreando el estado del sistema, en particular de las máquinas, e ir generando eventos en caso de que lo amerite. A modo de ejemplo, en el caso de que el módulo detecte una falla en alguna de las máquinas, generará un evento para actuar en consecuencia, por ejemplo, marcar la máquina como defectuosa y reasignar las tareas entre las máquinas no defectuosas. De igual forma que con el módulo *Scheduler*, todo esto se realiza aprovechando las primitivas provistas por BaSyx.

Finalmente, todas las máquinas que componen el sistema ejecutan su *Digital Twin* en un nuevo servidor dedicado, donde a cada virtualización de una máquina se le asocia una url mediante la cual se podrá acceder a una interfaz que BaSyx provee para consultar en tiempo real el estado de la misma.

Una vez desarrollado este sistema, se evaluaron distintos algoritmos de planificación para comparar su desempeño, así como el impacto en el sistema de dicho componente. El resultado de esta evaluación se presenta en la sección de experimentación (Sección 6) de este documento.

El objetivo principal es lograr desarrollar un sistema funcional que esté integrado con BaSyx y que funcione correctamente bajo un ambiente ficticio.

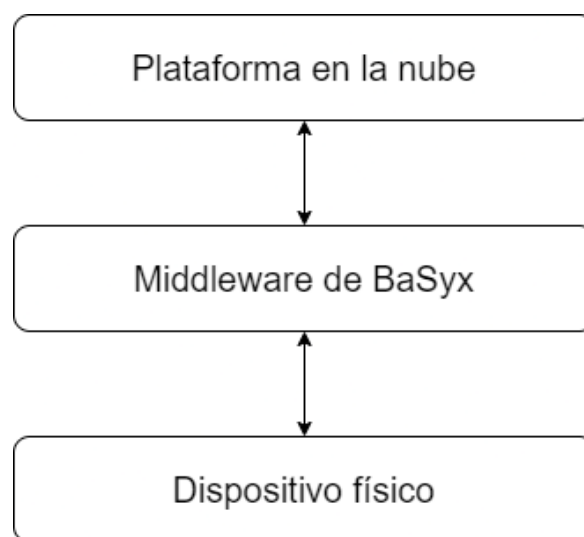
## 4. Arquitectura

Esta sección describe la arquitectura de la implementación realizada para desarrollar la prueba de concepto del uso de BaSyx en CMfg.

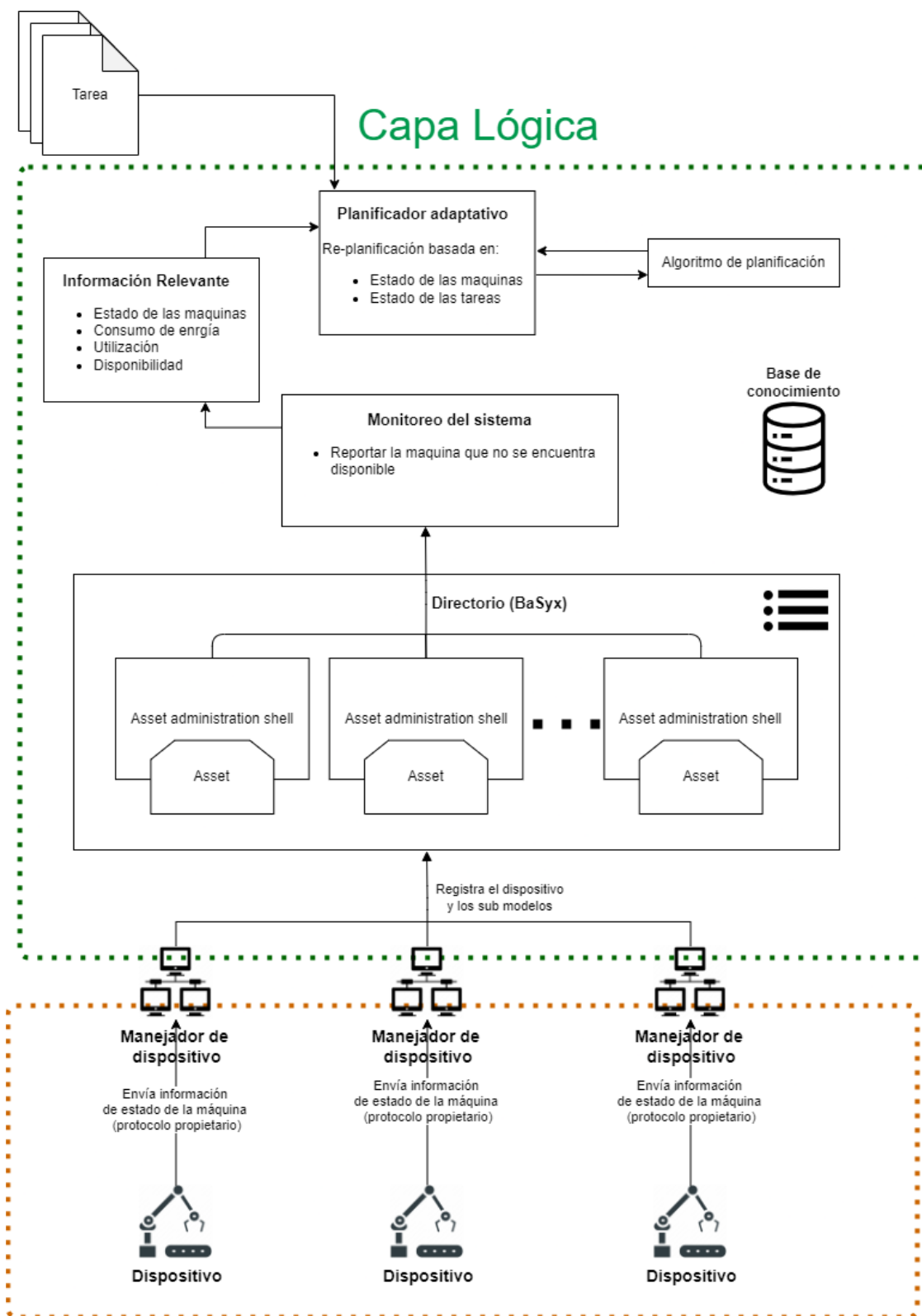
Para esta prueba de concepto se decidió diseñar un sistema de fabricación en la nube. En este sistema, todas las máquinas son impresoras 3D que pueden imprimir piezas en distintos materiales, tamaños y calidades, y donde tanto las máquinas como las órdenes (tareas) tienen una ubicación asociada. Con respecto a esto último, se decidió restringir la ubicación geográfica de estos elementos a Uruguay con el objetivo de facilitar la experimentación.

El sistema desarrollado consta de una aplicación web y un servidor. La aplicación web permite a diferentes usuarios crear nuevas máquinas y realizar órdenes de trabajo. El servidor se encarga de aceptar las órdenes, dividirlos en trabajos más simples y distribuir estos trabajos entre las distintas máquinas. Las máquinas ejecutan de forma independiente en sus servidores correspondientes, con el fin de mantenerlas monitoreadas.

La arquitectura del servidor está basada en la arquitectura planteada en [40]. Se decidió elegir esta arquitectura debido a su completitud, así como el hecho de que el uso de BaSyx podía complementar muy bien la arquitectura allí descrita (ver Figuras 10 y 11).



**Figura 10:** Capas que conforman un sistema de CMfg.  
Fuente: elaboración propia.



**Figura 11:** Diagrama de arquitectura de un sistema completo de fabricación en la nube  
Fuente: elaboración propia.

La arquitectura de la aplicación web consta de tres módulos principales. El primer módulo es el planificador (*scheduler*), que es el encargado de planificar, organizar y distribuir los diferentes pedidos con las distintas máquinas del sistema. Otro módulo es el verificador (*monitoring*), que se encarga de monitorear constantemente el estado de las máquinas en tiempo real, recopilando una gran cantidad de información de estas y notificando al *scheduler* en caso de que alguna de ellas falle o necesite especial atención.

Por último se cuenta con un módulo de BaSyx independiente. Es aquí donde la arquitectura implementada se separa de la utilizada en [40] y la razón de ser de esta prueba de concepto. El fin de BaSyx en esta arquitectura es el de funcionar como un *middleware* entre los dispositivos físicos (máquinas de una fábrica) con el sistema en la nube, haciendo uso de los *Digital Twins* que han sido nombrados en varias oportunidades a lo largo del documento.

#### 4.1 Planificador o *Scheduler*

Este módulo puede considerarse el núcleo del sistema, ya que es quien, a partir del algoritmo de *scheduler* implementado, se encarga de actuar en consecuencia a los distintos tipos de eventos que pueden aparecer en el sistema. Cada vez que se genera un evento de creación o rotura de máquinas o ingreso de una nueva orden de trabajo, el *scheduler* actuará inmediatamente para reorganizar las tareas actuales y distribuir la carga de trabajo de las nuevas órdenes entrantes.

Se le llama “orden” a un conjunto de trabajos simples que, en conjunto, forman un todo. Cuando llega una orden nueva, el *scheduler* la descompone en los distintos trabajos y los distribuye entre las distintas máquinas que fueron seleccionadas por el algoritmo implementado.

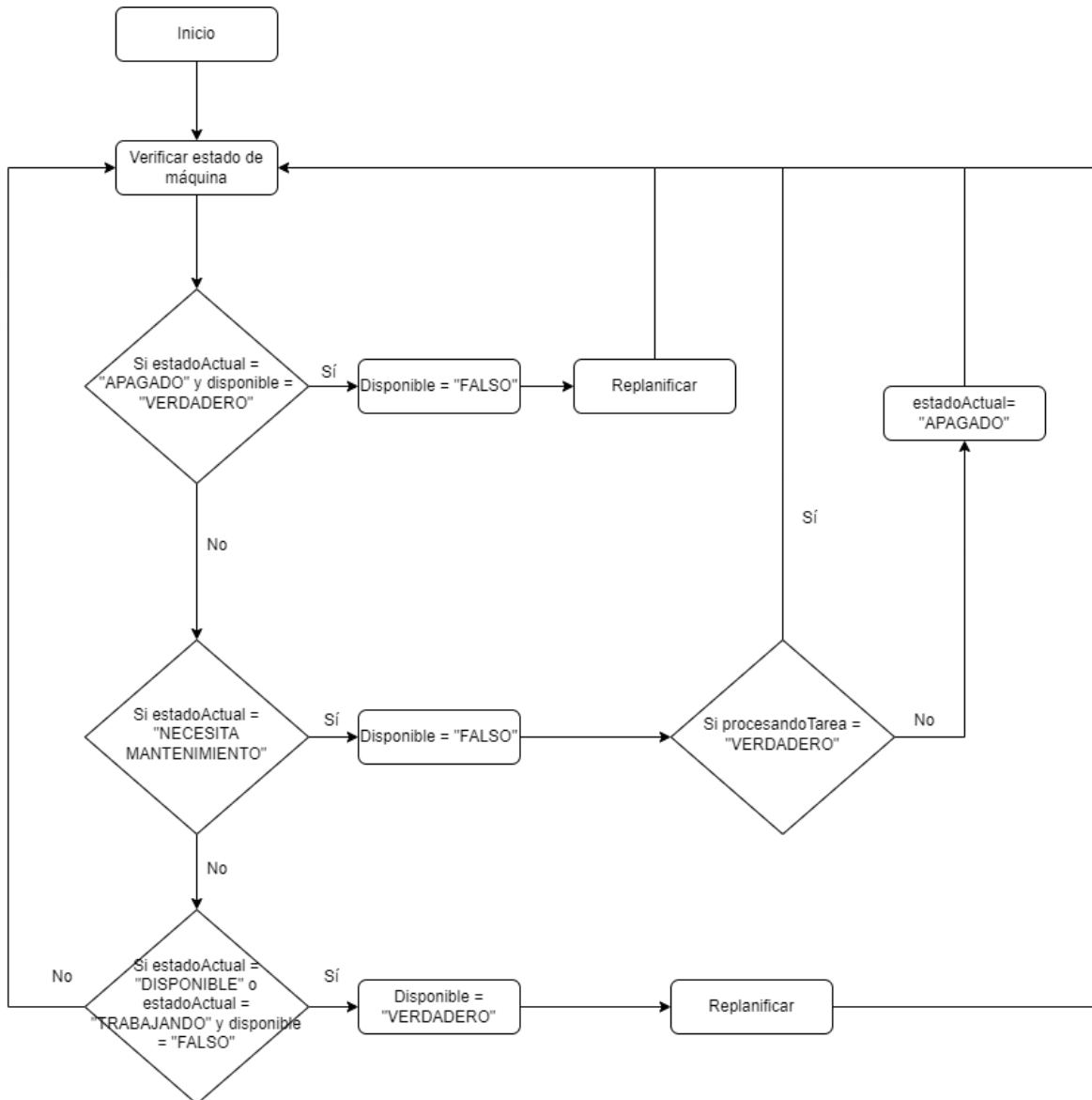
A su vez puede actuarse en consecuencia a eventos detectados por el *Monitoring*, módulo del que se hablará en detalle en la Sección 5.2. El *monitoring* es capaz de detectar cuando una máquina se encuentra en estado de falla para poder invocar al *scheduler* y reasignar el trabajo que la máquina defectuosa no puede realizar, entre las que se encuentran funcionando correctamente.

#### 4.2 Verificador o *Monitoring*

El módulo de verificación y monitoreo (*monitoring*) es quien se encarga principalmente del control de las máquinas. Es, además, quien más se beneficia del uso de BaSyx, puesto que se encuentra constantemente monitoreando las distintas máquinas del sistema y para ello es importante contar con una forma de comunicación con los dispositivos físicos.

Para realizar esta verificación, el *monitoring* se ejecuta en segundo plano como un proceso que se encuentra siempre vivo y que, cada cierta cantidad de tiempo, obtiene una gran cantidad de información de todas las máquinas del sistema. Esta información es procesada y, en caso de detectar que alguna de las máquinas no responde o necesita mantenimiento, le notifica inmediatamente al *scheduler*. Cuando el *scheduler* es notificado, el algoritmo de planificación reasigna las tareas que las máquinas defectuosas no pueden procesar.

De igual manera sucederá en el caso de que se detecte que una máquina que se encontraba rota o apagada ahora se encuentra funcionando correctamente. La Figura 12 a continuación muestra un diagrama con el algoritmo de verificación implementado para este sistema y que es detallado en mayor profundidad en la Sección 5.1.



**Figura 12:** Diagrama del algoritmo de verificación implementado.  
Fuente: elaboración propia.



El pseudocódigo que sigue, ilustra de otra forma el mismo algoritmo de verificación de la Figura 12:

Obtener todas las máquinas del sistema

Inicio bucle: para cada máquina del sistema

    Consultar el estado de la máquina

    Si la máquina se apagó

        Marcar máquina como no disponible

        Replanificar trabajos pendientes de la máquina

    Si la máquina necesita mantenimiento

        Verificar que la máquina no acepte nuevas tareas

        Si no hay tareas en cola

            Apagar máquina

    Si la máquina se recuperó de un incidente

        Marcar máquina como disponible

        Replanificar trabajos teniendo en cuenta esta máquina

Fin bucle: para cada máquina del sistema

### 4.3 Eclipse BaSyx

Finalmente, otra parte importante que compone el sistema es el módulo de BaSyx, una de las principales motivaciones para realizar esta implementación. BaSyx es una plataforma que brinda un gran conjunto de herramientas para la creación y administración de los *Digital Twins*, como también para su comunicación con los distintos dispositivos físicos.

En esta arquitectura se hace uso de BaSyx para la representación de los *Digital Twins* pero no así para su integración con dispositivos físicos, debido a que no se contaba con dispositivos que se ajustaran a la implementación que se deseaba hacer. Adicionalmente, el fin de esta prueba no era ir a tan bajo nivel, donde se trabajara directamente con el hardware.

El módulo de BaSyx en esta implementación, permite ejecutar un conjunto de *Digital Twins* en distintos servidores, y leer y modificar sus atributos a través de un conjunto de primitivas bien conocidas. De esta forma, es posible acceder a la información en tiempo real, así como también simular distintos estados y eventos que podrían suceder a un dispositivo físico en un caso real.

## 5. Implementación

En esta sección se detalla en mayor profundidad la implementación de la solución cuyo diseño se ha abordado en la sección precedente.

Siendo el principal objetivo de la implementación la realización de una prueba de concepto para validar los componentes que ofrece Eclipse BaSyx para plataformas de Industrias 4.0, se decidió instanciar una plataforma genérica en una red de impresoras 3D.

Para el desarrollo del prototipo del sistema de CMfg se eligió trabajar con el SDK<sup>9</sup> de BaSyx para .NET Core, por lo que el módulo del sistema que implementa los *Digital Twins* fue desarrollado sobre el framework .NET Core en su versión 3.1.

De la misma forma, el backend de la solución web que se comunica con el módulo anterior, también fue implementado sobre el mismo framework. La decisión responde únicamente al conocimiento y la comodidad de los desarrolladores para trabajar con el framework y, como ambos sistemas son independientes, no era necesario que fuesen construidos utilizando la misma tecnología.

Tanto la aplicación web como el módulo de *Digital Twins* desarrollados se comunican con la misma base de datos relacional, que fue construida sobre el motor de base de datos Microsoft SQL Server, en su versión 2019.

Finalmente, el frontend de la aplicación web fue desarrollado utilizando React.js y todo el sistema fue instalado y ejecutado en un equipo con sistema operativo Windows 10 Pro.

Adicionalmente, e independientemente de las implementaciones anteriores, se decidió desarrollar una aplicación de consola -estilo *daemon*- para generar y ejecutar casos de prueba. Esta experimentación es importante para validar las pruebas de concepto referentes a la utilización de diferentes estrategias de scheduling. En adelante, cuando el documento menciona el “simulador”, estará haciendo referencia a esta aplicación de consola estilo *daemon*.

### 5.1 Aplicación web

La aplicación web se divide en frontend, backend y persistencia de datos.

La capa de presentación de la aplicación permite que los usuarios interactúen con el sistema (la nube de CMfg) generando nuevas órdenes de trabajo y monitoreando en todo momento el estado del sistema.

Por simplicidad, en este desarrollo no es preciso que el usuario esté autenticado para utilizar el sistema, ya que el objetivo de esta prueba de concepto se centró en analizar las fortalezas y debilidades del uso de BaSyx para sistemas de fabricación en la nube. Se hace especial énfasis entonces, en que el desarrollo real de un sistema de CMfg deberá agregar

---

<sup>9</sup> Un software development kit (SDK) es un conjunto de herramientas de creación de software para una plataforma específica, que incluye los componentes básicos, depuradores y, por lo general, frameworks o librerías (SDK vs. API: What's the difference?, IBM, <https://www.ibm.com/cloud/blog/sdk-vs-api>)

controles de autenticación y permisos de los usuarios para poder utilizar cada componente de la aplicación.

La aplicación web desarrollada cuenta con un conjunto de vistas que permiten que el usuario interactúe con el sistema de CMfg. En primer lugar, se cuenta con una vista principal mediante la que el usuario puede visualizar las distintas máquinas que se encuentran en el sistema. Para cada máquina de la lista se podrá visualizar tanto su estado como una url que indica dónde se encuentra alojado su correspondiente *Digital Twin*. Adicionalmente, el sistema cuenta con una vista de detalles de máquina, que obtiene información directamente de su *Digital Twin*, de forma tal que la información que se muestra refleje la información real de la máquina en ese preciso momento.

Haciendo uso del menú de navegación ubicado en el lateral izquierdo del sitio, el usuario puede navegar a un conjunto de vistas.

Entre las vistas accesibles desde el menú de navegación se encuentran dos formularios para dar de alta nuevos elementos en el sistema. El primero es utilizado para crear nuevas máquinas, indicando sus características, mientras que el segundo permite crear nuevos órdenes de trabajo, que pueden estar compuestas por un conjunto de varias tareas.

Para finalizar, se consideró importante el poder realizar un seguimiento de las órdenes realizadas, para ello se implementó una nueva vista para la visualización de órdenes. Mediante esta vista, es posible visualizar las órdenes, así como también las distintas tareas que la componen y el estado de cada una de ellas. En esta vista, además, puede visualizarse una barra de progreso, que indica el estado de la orden y un tiempo estimativo de finalización de ésta.

La Figura 13 a continuación muestra una impresión de la pantalla principal del sistema web desarrollado.

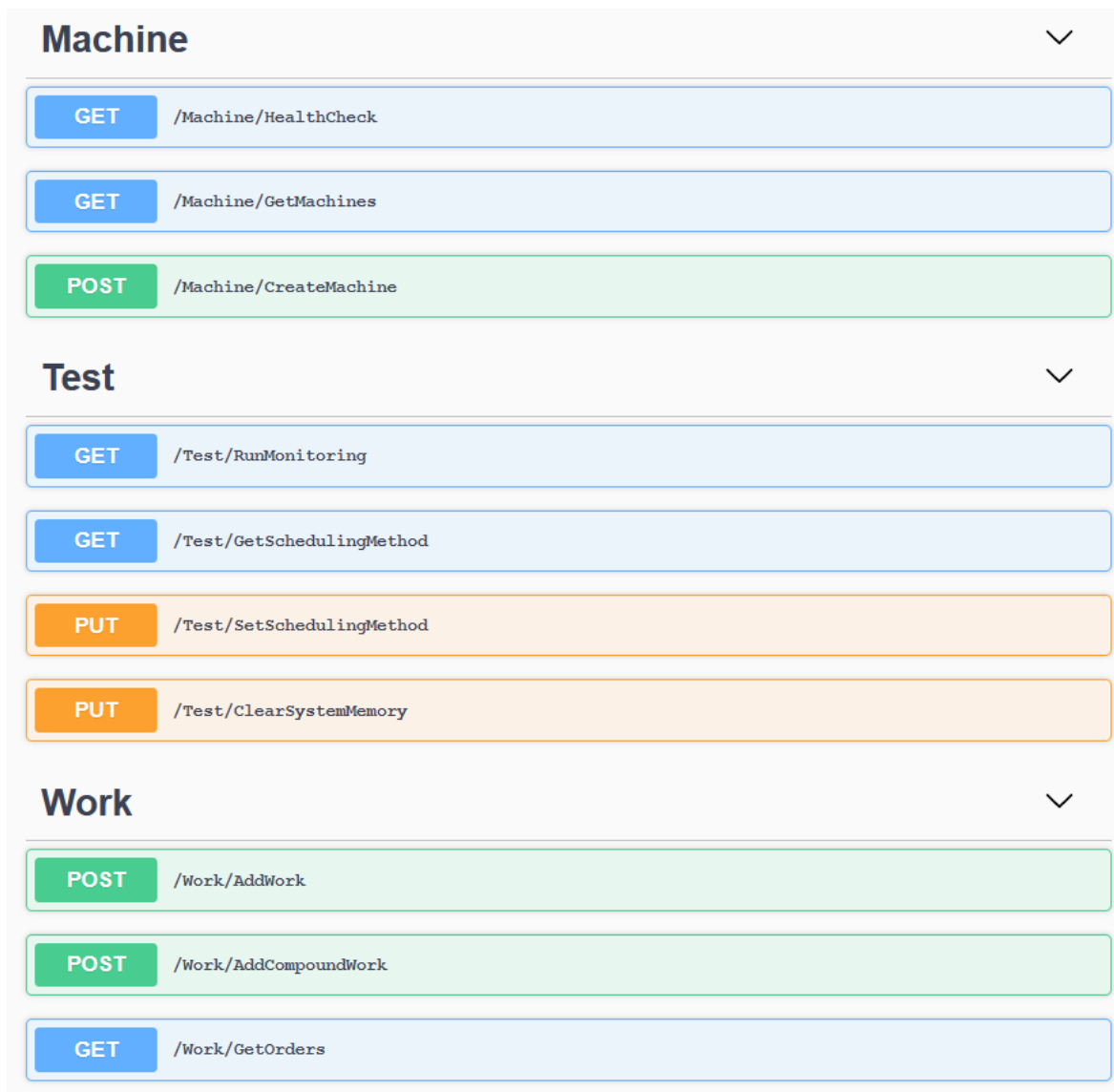
Name	URL	Actions
Machine_1	<a href="http://localhost:5000">http://localhost:5000</a>	<a href="#">Details</a>
Printer Machine For Port 5001	<a href="http://localhost:5001">http://localhost:5001</a>	<a href="#">Details</a>
Printer Machine For Port 5002	<a href="http://localhost:5002">http://localhost:5002</a>	<a href="#">Details</a>
Printer Machine For Port 5003	<a href="http://localhost:5003">http://localhost:5003</a>	<a href="#">Details</a>
Printer Machine For Port 5004	<a href="http://localhost:5004">http://localhost:5004</a>	<a href="#">Details</a>

**Figura 13:** Interfaz web de una de las pantallas de la aplicación que muestra el listado de máquinas del sistema y permite acceder a sus detalles.

Fuente: elaboración propia.

El backend de la aplicación consta de una Rest API con un controlador para los casos de uso concernientes a las máquinas que forman parte del sistema y otro controlador para los casos de uso referentes a las órdenes. Nuevamente, se optó por implementar una API por simplicidad. Para una plataforma de fabricación en la nube de acceso masivo, otras arquitecturas de software diferentes podrían ser evaluadas. Una arquitectura de microservicios, por ejemplo, permite escalar con mayor facilidad que una única Rest API.

Los endpoints y los modelos de datos utilizados por la API fueron documentados usando Swagger (Figura 14) y dicha información puede leerse accediendo a la URL base de la API, utilizando cualquier navegador web



**Figura 14:** Página de Swagger con la documentación de los endpoints y modelos utilizados por la API.

Fuente: elaboración propia.

Además de los controladores y repositorios referentes a las máquinas y órdenes de trabajo, la API incluye también un módulo planificador (*scheduler*) y un módulo de servicio verificador (*monitoring service*) que son parte del núcleo más importante del sistema de CMfg.

El *scheduler* resulta fundamental, ya que su ejecución está presente en prácticamente todas las acciones realizadas por el sistema. Cada vez que una nueva orden de trabajo es generada, es dividida en pequeñas tareas que serán distribuidas de forma conveniente entre las diferentes máquinas por el algoritmo de scheduling. Asimismo, cuando se detecta que una máquina deja de estar disponible (por mantenimiento planificado, rotura) o cuando se detecta una nueva máquina disponible, el scheduler vuelve a tomar relevancia. El *scheduler* se encarga de realizar la reasignación de trabajo entre las máquinas disponibles, según corresponda.

La literatura analizada durante el relevamiento del estado del arte de CMfg [Anexo I] muestra que la selección de un algoritmo de planificación eficiente es un problema en sí mismo y el algoritmo seleccionado impactará directamente en la eficiencia del sistema. Teniendo esto en cuenta y con el objetivo de mostrar que la utilización de BaSyx no limita la selección y el uso de determinado algoritmo de planificación, se decidió implementar dos algoritmos diferentes. El módulo de scheduler tiene entonces un algoritmo simple FIFO y un algoritmo con cierta complejidad, que se decidió nombrar LESS WORKLOAD y se ilustra en la Figura 15.

---

**Algoritmo desarrollado:** el algoritmo LESS WORKLOAD inicia con la llegada de una nueva tarea que deberá ser procesada por el sistema para cumplir con la orden de un cliente.

Una vez que arriba la nueva tarea al sistema, el algoritmo filtra de un listado de máquinas que mantiene en memoria aquellas máquinas que son capaces de satisfacer el pedido, esto es, realizar la tarea. Para ello, el algoritmo tiene en cuenta la disponibilidad de la máquina (es decir, que no se encuentre apagada o en reparación) y la compatibilidad con la tarea (esto es, que la máquina soporte el tamaño de pieza y el material de impresión solicitados).

Una vez realizado este filtrado, el listado de máquinas resultante es ordenado por proximidad geográfica entre la ubicación de la máquina y la ubicación de la orden y se toman en cuenta las 100<sup>10</sup> máquinas geográficamente más cercanas.

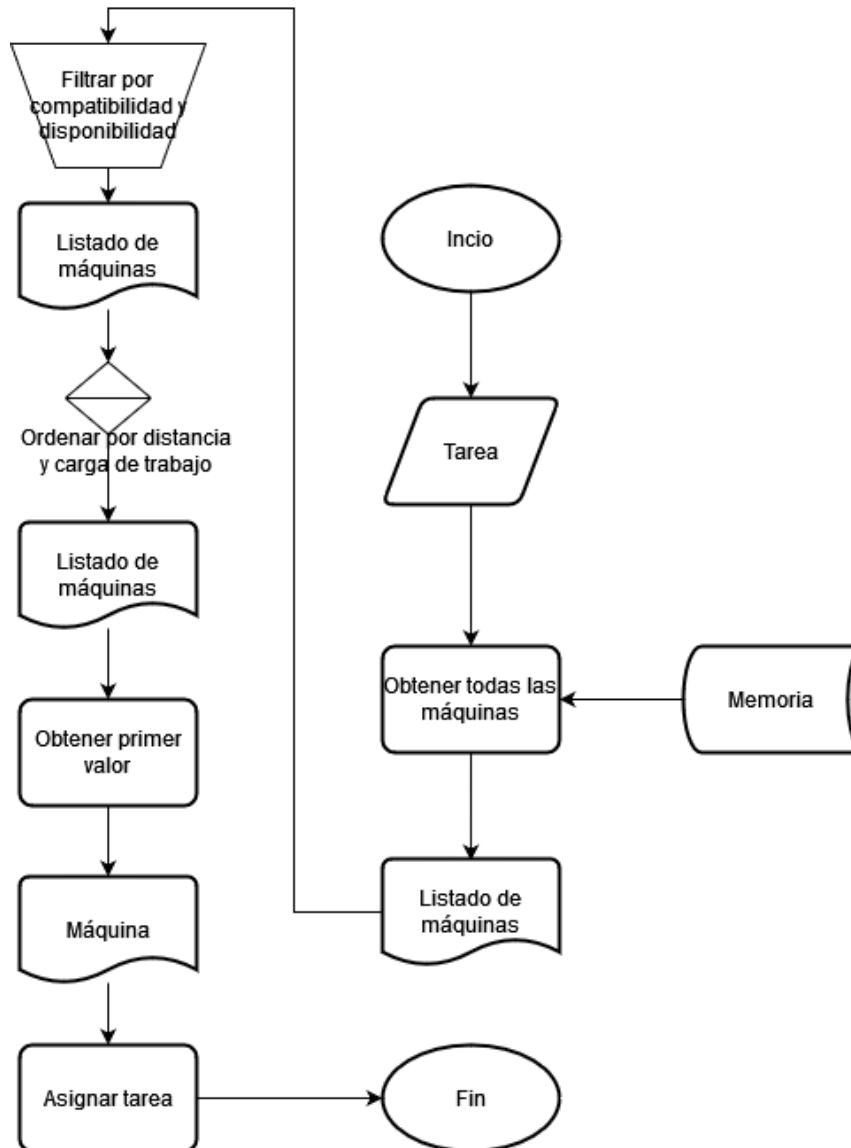
Finalmente, el algoritmo selecciona de entre las 100 máquinas más cercanas a la ubicación de la orden, aquella máquina que tiene la menor carga de trabajo, con el objetivo de que la tarea se realice lo antes posible.

Con esta máquina seleccionada, se procede a realizar la vinculación entre tarea y máquina y finaliza el caso de uso.

---

---

<sup>10</sup> La cantidad de máquinas tenidas en cuenta puede modificarse de acuerdo a los requerimientos del sistema. El número actual fue seleccionado convenientemente a efectos de ilustrar la importancia de la cercanía geográfica entre la ubicación de la orden y la ubicación de la máquina. En escenarios reales donde el costo del traslado de las piezas es no despreciable, puede realizarse este filtrado teniendo en cuenta una distancia máxima en lugar de las X máquinas geográficamente más cercanas o utilizarse otro tipo de reglas aún más complejas.



**Figura 15:** Diagrama que ilustra el algoritmo de planificación “LESS WORKLOAD” implementado  
Fuente: elaboración propia.

En este sentido, la API provee un endpoint que habilita la capacidad de migrar de un algoritmo a otro a conveniencia. Cuando el algoritmo FIFO se encuentra habilitado, el sistema asignará cada tarea que compone una orden a aquella máquina que se encuentra geográficamente más cerca de las coordenadas de la orden de entre las máquinas que son capaces de satisfacer el pedido (es decir, soportan el tipo de material, el tamaño de la pieza y la calidad de la misma). Cuando LESS WORKLOAD se encuentra habilitado, las tareas que componen una orden serán enviadas a las máquinas que tengan menor carga de trabajo, de entre las 100 máquinas geográficamente más cercanas que sean capaces de satisfacer el pedido (según material, tamaño y calidad de la pieza). El objetivo de este algoritmo es mejorar, en la medida de lo posible, el tiempo de finalización de las órdenes, a

cambio de posiblemente aumentar las distancias entre las máquinas que trabajarán sobre las diferentes tareas que componen la orden.

Los resultados relativos entre ambos algoritmos serán detallados en la sección de experimentación de este documento.

El segundo módulo del sistema que fue nombrado luego del *scheduler* es el verificador (o *monitoring*). El objetivo de este módulo es controlar en tiempo real el estado de las máquinas que componen la nube y la implementación de este módulo fue realizada como un proceso de fondo o en segundo plano (*background service*). Como rutina asincrónica que se ejecuta en segundo plano, este servicio controla el estado de cada una de las máquinas del sistema y se comunica directamente con el *digital twin* de BaSyx.

Cuando el verificador encuentra un cambio en el estado de una máquina, entonces ejecuta la rutina correspondiente para evitar, en la medida de lo posible, la interrupción del sistema. El objetivo último es que las caídas de las máquinas (por roturas, mantenimiento o cualquier otro motivo) sean transparentes a los clientes.

La API es, además, la encargada de integrar los diferentes *Digital Twins* de BaSyx para formar un único sistema de fabricación en la nube.

De acuerdo a lo leído durante la revisión sistemática, se considera que un sistema real de fabricación en la nube cooperativo implemente algunos controles y restricciones extras. Un sistema real debería, por ejemplo, asegurarle a los productores (que ponen a disposición sus máquinas) que los trabajos serán repartidos de forma “solidaria” entre todos los productores, esto es, que no existirá un productor al que nunca se le asigne ninguna tarea.

La sección de persistencia de datos será abordada en 5.3.

## 5.2 Digital Twins de BaSyx

El módulo de *Digital Twins* de BaSyx consiste en un servicio que simula máquinas de impresión 3D.

Tal y como fue presentado en la Sección 3, Eclipse BaSyx ofrece una implementación de referencia que permite abstraer las características de las máquinas físicas y simularlas en software.

La implementación de este módulo fue desarrollada completamente desacoplada de la aplicación web de la sección 5.1. De esta forma, se muestra que la administración del sistema, incluyendo la recepción y distribución de tareas entre las máquinas, puede implementarse en un servidor, mientras la ejecución de los *Digital Twins* de las máquinas físicas puede ejecutar en otro completamente diferente. Adicionalmente, pueden utilizarse diferentes plataformas o lenguajes de programación, ya que la integración es transparente para el usuario.

Para el desarrollo de esta prueba de concepto, se decidió simular máquinas de impresión 3D con características tales como material soportado (ver Tabla 1), tamaños soportados (ver



Tabla 2), calidades de impresión soportadas (ver Tabla 3) y ubicación geográfica de la máquina.

La utilización de los AAS Sub Models de BaSyx (presentados en la sección 3.1 de este documento) en este caso facilitó la implementación de operaciones y estados sobre un *Digital Twin*. El SDK de BaSyx provee una clase fundamental para lograr este objetivo, llamada *SubmodelElement*, que permite estructurar un AAS representando un dominio específico o un tema bien definido.

Para cada submodelo de BaSyx, es posible definir un conjunto de elementos referenciados así como también un conjunto de operaciones y eventos que permiten interactuar con el *Digital Twin*.

Una vez que un usuario desea dar de alta una nueva máquina en el sistema, un nuevo AAS es creado con la información provista por el usuario. Estos AAS encapsulan un *SubmodelElement* que expone un conjunto de operaciones para así poder:

- Leer el estado de la máquina
- Obtener cada uno de los valores de sus atributos
- Agregar nuevos trabajos a la cola de trabajos
- Quitar uno o más trabajos de la cola de trabajos
- Obtener la carga de trabajo actual para la máquina
- Obtener la fecha estimada de finalización de las tareas de la máquina
- Una operación extra, que será utilizada durante la experimentación (Sección 6), para modificar manualmente el estado de la máquina.

Además, cada uno de estos AAS son ejecutados en un servidor independiente, al cual es posible acceder mediante una dirección configurable en el código.

En este proyecto en particular, la dirección donde se encuentran alojados los *Digital Twins*, es de la forma <http://localhost:XXXX>, donde XXXX es el número de puerto que inicia en 5000 para la primera máquina y va incrementándose de uno en uno para las subsiguientes máquinas.

Material soportado
PLA_FILAMENT
ABS_FILAMENT
PETG_FILAMENT
POLYAMIDE_POWDER
ALUMINA_POWDER
RESIN

**Tabla 1:** listado de materiales soportados por las máquinas

Tamaños soportados
SMALL
MEDIUM
LARGE
EXTRA_LARGE

**Tabla 2:** listado de tamaños soportados por las máquinas

Todas estas operaciones soportadas por los *Digital Twins*, se encuentran debidamente documentadas en formato Swagger para cada uno de los *Digital Twins*. Para acceder a la documentación, basta con acceder al puerto que haya sido configurado para esto. En el caso de esta implementación, es el puerto 10000.

Calidades soportadas
LOW
HIGH

**Tabla 3:** listado de calidades soportadas por las máquinas

Estados posibles
OFFLINE
AVAILABLE
WORKING
NEEDS_MAINTENANCE

**Tabla 4:** listado de los posibles estados en los que puede encontrarse una máquina

Cada uno de los *Digital Twins* implementados para las máquinas de impresión 3D de este sistema es independiente de los demás, pero tiene la misma estructura general.

Cada máquina puede soportar un único material de impresión, mientras que puede soportar entre 1 y N tamaños y calidades. Asimismo, durante el desarrollo de este sistema, las máquinas se asumen perfectas -es decir, que a excepción de cuando la máquina se rompe o apaga, el tiempo que dura la impresión de las piezas no se verá afectado-.

Finalmente, cada uno de los *Digital Twins* cuenta con un proceso en segundo plano que simula la ejecución de las tareas y va despachándolas a medida que pasa el tiempo.

Una de las ventajas de la fabricación comunitaria en la nube es la posibilidad de recabar grandes cantidades de información (big data). Esto permitiría usarla como entrada en procesos de, por ejemplo, aprendizaje automático, para aplicar procesos de mejora continua. Se llegaría, incluso, a prevenir fallas en las máquinas y a planificar el mantenimiento de las mismas, mejorando los tiempos de inactividad.

Por simplicidad, la implementación realizada en esta oportunidad, no incluye el subsistema de recopilación y procesamiento de datos, sin embargo, resulta importante realizar el comentario anterior.

Vinculado con lo expuesto en el párrafo anterior, los DT desarrollados ofrecen un endpoint para modificar el estado de la máquina para mantenimiento predictivo de la misma. Este endpoint será consumido por el simulador para justamente, simular, la necesidad de algunas máquinas de detener la operativa durante cierto tiempo para la realización de tareas de limpieza, ajuste y mantenimiento. La lista de estados posibles puede verse en Tabla 4.

### 5.3 Base de datos

Tanto la aplicación web de la Sección 5.1 como la implementación de los *Digital Twins* de BaSyx de la Sección 5.2 utilizan una base de datos para persistir información relevante.

La elección de utilizar una única base de datos relacional para esta implementación, se apoya fuertemente en realizar simplificaciones de las arquitecturas de software relevadas durante la revisión de literatura del estado del arte de CMfg. Para el objetivo de este trabajo, que consiste en realizar una prueba de concepto de arquitectura de CMfg y mostrar el uso de Eclipse BaSyx, la utilización de una base de datos compartida entre los *Digital Twins* (5.2) y el sistema central (5.1) resulta más que suficiente.

Implementaciones más avanzadas y realistas podrían apoyarse en la utilización de bases de datos distribuidas, con redundancia de datos, o bases de datos no relacionales para mejorar la eficiencia de los procesos, para reducir latencias y para reducir los riesgos.

El sistema desarrollado hace uso de la base de datos para persistir toda aquella información relevante para el correcto funcionamiento del mismo. Realiza copias de seguridad de las máquinas que se encuentran disponibles en el sistema, incluyendo todas sus características y su carga de trabajo en todo momento.

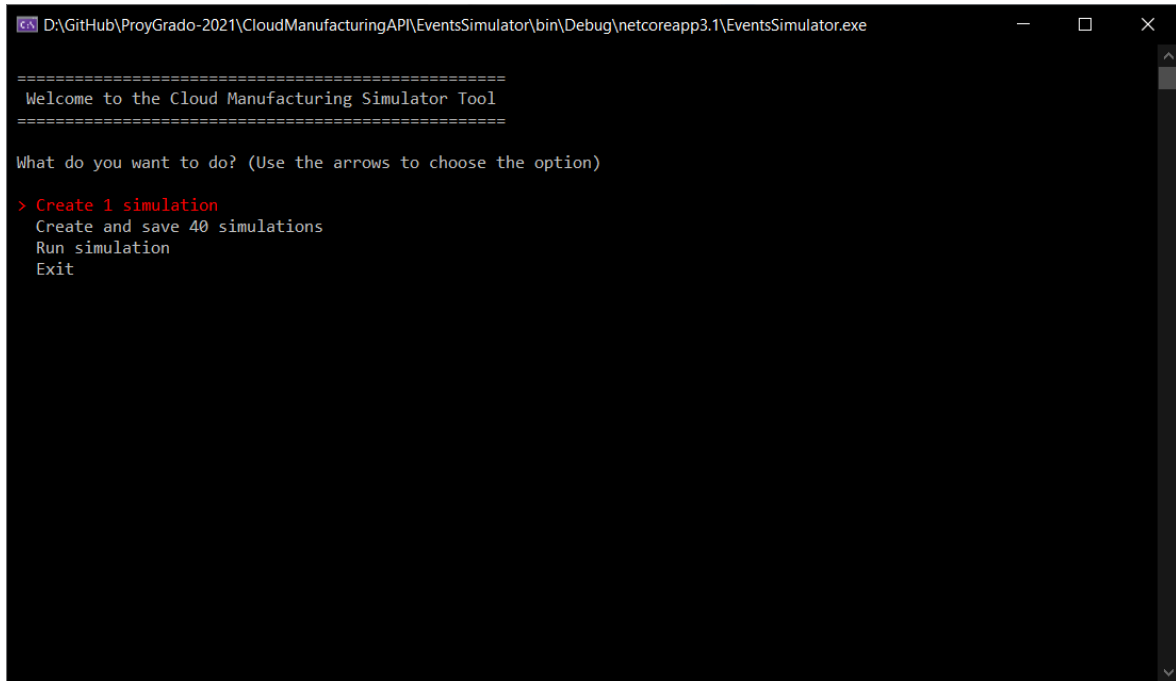
Con el objetivo de centralizar las operaciones sobre la base de datos, se desarrolló una librería de .NET Core -CloudManufacturingDBAccess- que provee métodos de lectura y escritura sobre las tablas correspondientes. Esta librería mantiene, también, el control de migraciones sobre el *schema* de la base de datos.

Tanto la aplicación web como el servicio de *Digital Twins*, hacen uso de esta librería para comunicarse con la base de datos.

### 5.4 Diseño experimental

Una herramienta extra que se decidió implementar luego de construido el sistema de CMfg inicial que incluía una aplicación web y un módulo de virtualización de máquinas, es una herramienta para ejecutar casos experimentales (presentado al comienzo de esta sección).

Esta herramienta consiste en una aplicación de consola que ofrece, básicamente, dos tipos de casos de uso, por un lado permite generar archivos de pruebas y por el otro permite ejecutar estas pruebas previamente generadas. La interfaz de usuario consta de un menú de consola, como ilustra la Figura 16

A screenshot of a Windows command prompt window titled "D:\GitHub\ProyGrado-2021\CloudManufacturingAPI\EventsSimulator\bin\Debug\netcoreapp3.1\EventsSimulator.exe". The window displays a text-based interface for the "Cloud Manufacturing Simulator Tool". It starts with a welcome message and a prompt asking the user what they want to do. The user has entered the command "> Create 1 simulation", which has triggered a list of options: "Create and save 40 simulations", "Run simulation", and "Exit".

```
D:\GitHub\ProyGrado-2021\CloudManufacturingAPI\EventsSimulator\bin\Debug\netcoreapp3.1\EventsSimulator.exe

=====
Welcome to the Cloud Manufacturing Simulator Tool
=====

What do you want to do? (Use the arrows to choose the option)

> Create 1 simulation
  Create and save 40 simulations
  Run simulation
  Exit
```

**Figura 16:** Interfaz de usuario de la herramienta de experimentación, nombrada “simulador”.

Fuente: elaboración propia.

La herramienta de experimentación, en adelante llamada “simulador”, permite crear una prueba única o un lote de 40 pruebas independientes, así como permite también ejecutar un archivo de pruebas en particular o todos los archivos de pruebas disponibles.

Una experimentación o prueba en particular, consiste en una serie de eventos que tienen lugar a lo largo de cierta cantidad de unidades de tiempo fijas. En cada instante de tiempo de la prueba, el algoritmo de creación de casos de prueba seleccionará, con una probabilidad predefinida, un evento de la lista de eventos soportados (ver Tabla 5).

Las probabilidades de cada evento se detallan en la Sección 6 (Experimentación) de este documento.

Los datos generados por el simulador son persistidos en archivos de texto que serán leídos por el programa al momento de decidir realizar la experimentación.

Eventos de la experimentación
BREAK_MACHINE
CREATE_MACHINE
NEW_ORDER
NOTHING

**Tabla 5:** eventos soportados por el simulador

Cuando el usuario se disponga a ejecutar algún caso de prueba, el simulador le ofrecerá la posibilidad de seleccionar un conjunto de eventos previamente generado o la posibilidad de ejecutar todos los archivos de pruebas disponibles. El sistema, entonces, comenzará a ejecutar uno a uno todos los eventos que tienen lugar en el archivo de experimentación seleccionado, comunicándose directamente con la Rest API de la Sección 5.1 o con el *Digital Twin* correspondiente de la Sección 5.2 de este documento.

---

**Precondición:** para que las pruebas se desarrollen de forma correcta, tanto el *backend* de la aplicación web como el módulo de *Digital Twin* deben encontrarse en ejecución y accesibles para el simulador.

---

Una vez ejecutados todos los eventos de la prueba, el sistema se mantendrá en espera activa hasta que todos los trabajos hayan sido procesados completamente por el sistema. Luego guardará de forma segura para su posterior análisis los resultados de la experimentación (que son persistidos en tiempo real por los *Digital Twins*) en un archivo que no volverá a ser sobrescrito y finalizará la prueba.

Los archivos con los resultados de las pruebas serán oportunamente analizados al final de la Sección 6 de este documento.

## 6. Experimentación

La Sección 6 desarrollará todo lo referente a experimentación, desde la generación de casos de prueba, hasta el análisis de los resultados obtenidos, pasando por la ejecución de los diferentes archivos con datos de prueba.

La primera versión de la implementación del sistema fue pensada para simular de forma lo más certera posible las características de un entorno de CMfg real. Sin embargo, para la realización de los experimentos, fue necesario realizar algunos ajustes en el sistema. Con el fin de evitar tiempos de ejecución muy extensos durante la experimentación -teniendo en cuenta que en “tiempo real” las impresiones 3D podrían tardar varias horas en finalizar-, los casos de prueba se diseñaron con base en una cantidad finita y discreta de unidades de tiempo. En cada una de estas unidades de tiempo tiene lugar un evento determinado, como fue comentado previamente en la Sección 5.4.

Surgió, entonces, la necesidad de modificar el diseño y la implementación inicial de varias partes del sistema para que comiencen a funcionar sobre la base de lo que llamamos unidades ficticias de tiempo, en lugar del tiempo real y continuo con el que trabajaban previamente.

Las principales modificaciones tuvieron lugar en el código de simulación de trabajo en los *Digital Twins*, y en el servicio de verificación (*monitoring service*). Este último, pasó de ser un servicio que ejecutaba constantemente en segundo plano a ejecutarse mediante la invocación de un *endpoint*. Este endpoint fue específicamente diseñado para que el simulador lo consuma cada cierta cantidad de unidades de tiempo ficticias. Una vez normalizado el funcionamiento completo del sistema para soportar las ejecuciones basadas en “unidades ficticias” de tiempo discreto, se procedió a realizar los experimentos propiamente dichos.

Con estos experimentos, además de corroborar el correcto funcionamiento del sistema bajo distintos tipos de situaciones, se busca evaluar el desempeño del *scheduler* implementado. Para ello fueron realizadas un total de 120 pruebas independientes, haciendo uso de dos *schedulers* diferentes: los algoritmos FIFO y LESS\_WORKLOAD que fueron oportunamente explicados en la Sección 5.1.

Del total de 120 ejecuciones de casos de prueba, 60 de ellos tuvieron lugar sobre el *scheduler* original (LESS\_WORKLOAD) y los restantes 60 mediante el *scheduler* FIFO. Para que los resultados fueran comparables, fueron utilizados los mismos 60 casos de prueba con ambos *schedulers*.

Los detalles de los casos de prueba se desarrollan en la Sección 7.1. De momento, basta con saber que cada uno de estos archivos cuenta con un total de 120 eventos independientes que son ejecutados de forma sincrónica por el simulador.

## 6.1 Casos de prueba

Para la creación de casos de pruebas se decidió definir, en primer lugar, con qué cantidad de unidades de tiempo trabajar. En este sentido, como se adelantó en la introducción de esta Sección, se eligió trabajar partiendo de casos de prueba con 120 unidades de tiempo, donde en cada una de ellas tendría lugar un evento de los mencionados en Tabla 5.

Además de definir la cantidad total de unidades de tiempo que tendría cada uno de los casos de prueba, también se definió la cadencia<sup>11</sup> con la cual sería forzada la ejecución del verificador (la justificación de esto se encuentra al comienzo de la Sección 6), definiendo que el verificador se ejecute cada 5 unidades de tiempo ficticio. La ejecución del verificador tiene un rol fundamental al momento de detectar máquinas que se rompen, apagan o necesitan mantenimiento. También detecta máquinas que se recuperan y vuelven a estar disponibles luego de haber sufrido uno de estos inconvenientes, y máquinas nuevas que son creadas por el sistema y deben comenzar a ser tenidas en cuenta, para que el *scheduler* reaccione en consecuencia.

Con las bases ya definidas, solo resta definir las probabilidades que se le asignarán a cada evento.

Con base en la literatura estudiada durante el relevamiento del estado del arte, se decidió que las probabilidades de creación y necesidad de mantenimiento de las máquinas debía variar en el tiempo. Al comienzo del tiempo, la probabilidad de que nuevas máquinas ingresen al sistema es mayor, y a medida que el tiempo avanza, aumenta también la probabilidad de que una máquina necesite mantenimiento y deba quedar fuera de servicio. Luego, para simplificar este modelo, se definió que esta variación de probabilidad para ambos eventos tuviese lugar en el primer y último cuarto del tiempo (creación y mantenimiento de máquinas, respectivamente).

Finalmente, se ejecutaron las mismas pruebas sobre dos sistemas de CMfg similares que solamente difieren en el algoritmo de planificación implementado. Esto, con el objetivo de demostrar el rol fundamental del módulo de *scheduling* en sistemas de fabricación en la nube, además de realizar la prueba de concepto del uso de BaSyx.

Por esta razón, fueron definidos dos conjuntos de probabilidades de sucesos, para que la comparación entre ambos sistemas se realizara sobre una base de 60 pruebas independientes. 30 de estas pruebas utilizan un conjunto de probabilidades y 30 utilizan otro conjunto diferente de probabilidades.

---

<sup>11</sup> Cadencia: ritmo o repetición de determinados fenómenos, como sonidos o movimientos, que se suceden con cierta regularidad. (Fuente: Diccionario de la lengua española, Real Academia Española)



Evento	Probabilidad 1	Probabilidad 2
CREATE_MACHINE_HIGH	30	30
CREATE_MACHINE	10	10
BREAK_MACHINE	10	15
BREAK_MACHINE_HIGH	20	25
NEW_ORDER	100	100
NOTHING	100	75

**Tabla 6:** Distribución de probabilidades utilizadas para los eventos al momento de generar los dos conjuntos de datos de las pruebas.

---

**Precondición:** cada una de las pruebas será ejecutada sobre un sistema limpio, con una base de datos vacía que será inicializada con un conjunto reducido de máquinas que simularán el estado inicial de un sistema de CMfg. Este estado inicial consta de una máquina que soporte todos los tamaños de pieza y todas las calidades por cada material de impresión 3D.

---

El proceso de creación de un caso de prueba consiste en sortear con equiprobabilidad un número entre 0 y 100 para cada unidad de tiempo que dura la prueba. Con base en el resultado de esta operación, se selecciona el evento que tendrá lugar en el instante dado, de la siguiente manera. Siendo  $X$  el valor sorteado, entonces se construye una lista con todos los elementos cuya probabilidad asignada en Tabla 6 supere el valor  $X$ <sup>12</sup>, luego, se selecciona con equiprobabilidad uno de estos eventos, que será el evento que finalmente tendrá lugar en el instante de tiempo que se está trabajando.

El proceso se repite para cada uno de las 120 unidades de tiempo.

Cuando el evento seleccionado es “NEW\_ORDER”, tiene lugar un nuevo sorteo de probabilidad. Para las pruebas, se decidió que las órdenes podrán estar compuestas por entre 1 y 10 tareas, por lo que cada vez que una nueva orden tiene lugar, se sortea con equiprobabilidad la cantidad de tareas que la componen.

Los datos de las pruebas son persistidos en archivos de texto que serán leídos por el simulador cuando el usuario selecciona la opción de ejecutarlos.

---

<sup>12</sup> Para los eventos de creación de máquinas y de necesidad de mantenimiento de las mismas, la probabilidad a tener en cuenta depende de para qué unidad de tiempo se está sorteando la probabilidad: si se trata del primer cuarto de unidades de tiempo, entonces se tiene en cuenta la probabilidad CREATE\_MACHINE\_HIGH y BREAK\_MACHINE; si se trata del último cuarto de unidades de tiempo, se tiene en cuenta las probabilidades CREATE\_MACHINE y BREAK\_MACHINE\_HIGH; finalmente, si se trata de unidades intermedias de tiempo, se tiene en cuenta las probabilidades CREATE\_MACHINE y BREAK\_MACHINE.

Al momento de ejecutar las pruebas, el simulador leerá los archivos de a uno e iterará sobre los 120 eventos que contienen, realizando las invocaciones correspondientes a la API de la Sección 5.1 o al *Digital Twin* correspondiente de la Sección 5.2.

Una vez ejecutados los 120 eventos de la prueba, el simulador se mantiene en espera activa hasta que todas las órdenes hayan sido procesadas por los *Digital Twins* correspondientes. Luego de ello, se encarga de reiniciar la base de datos, limpiar la memoria de la API, eliminar los *Digital Twins* y preparar el siguiente caso de prueba.

## 6.2 Resultados obtenidos

Este conjunto de pruebas, como ya fue mencionado en secciones anteriores, tenía como fin demostrar el correcto funcionamiento de la solución de fabricación en la nube con el uso de BaSyx. También era de interés demostrar la importancia de los *schedulers* y cómo estos pueden impactar significativamente en el desempeño del sistema.

Para ello se decidió tomar en consideración dos métricas principales: en primer lugar, el tiempo promedio de finalización de una orden para cada uno de los *schedulers* y en segundo lugar, los extremos de los tiempos -más cortos y más extensos- de finalización de esta. Estos valores permiten corroborar que el total de órdenes despachadas finalizan correctamente y, además, comparar el rendimiento de los dos algoritmos de planificación implementados.

Es de suma importancia tener en cuenta que todas las pruebas realizadas para este caso experimental fueron ejecutadas bajo un conjunto de unidades ficticias definidas por el equipo. Por lo tanto todos los resultados que serán mostrados en esta sección y en la siguiente estarán representados en estas unidades y no reflejarán al 100% el valor real. Aún así, se considera que estos resultados son más que suficiente para poder realizar las comparaciones pertinentes y obtener resultados conceptualmente correctos.

El primer lote de 30 ejecuciones realizado fue utilizando el conjunto de probabilidades 1 (ver Tabla 6) y se ejecutó para ambos *schedulers*. Con el *scheduler* FIFO, se obtuvo un tiempo promedio de finalización de órdenes de 177.64 unidades de tiempo ficticio (en adelante, FTU<sup>13</sup>), alcanzando el valor más alto de 622.82 FTU y el valor más bajo en 9.027 FTU. En cuanto al *scheduler* LESS\_WORKLOAD, este logró un tiempo promedio de finalización de orden de 131.775 FTU, con un máximo de 585.87 FTU y un mínimo de 7.565 FTU.

El siguiente lote de 30 ejecuciones fue realizado con casos de prueba generados utilizando el conjunto de probabilidades 2 (Tabla 6). Con este lote, para el algoritmo FIFO se obtuvo un tiempo promedio de 207.51 FTU, con máximo y mínimo de 633.69 FTU y 7.565 FTU respectivamente. Para concluir, los resultados para el algoritmo LESS\_WORKLOAD fueron de un tiempo promedio de 161.87 FTU, con un máximo de 612.641 FTU y un mínimo de 5.112 FTU.

Los valores obtenidos podrán encontrarse en las Figuras 16 y 17,

---

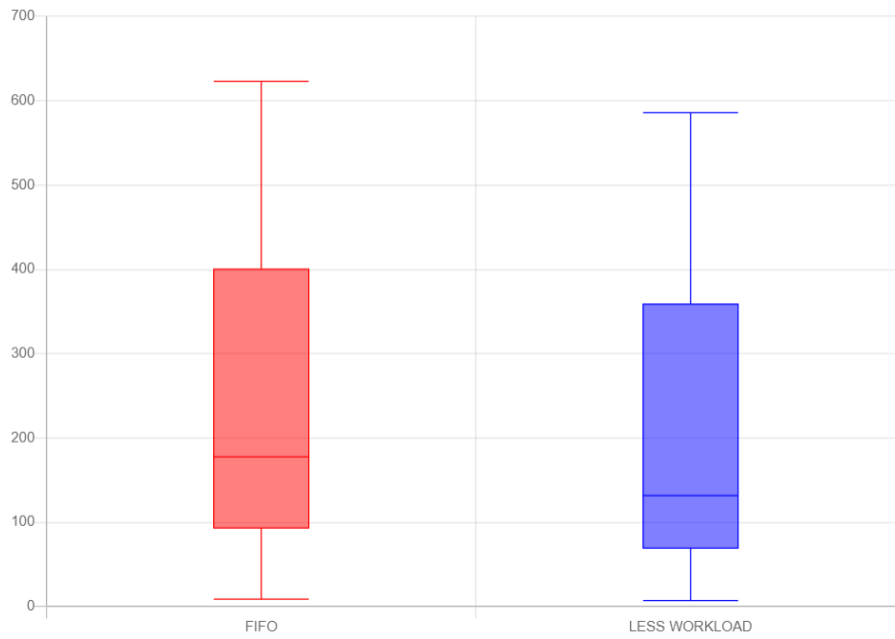
<sup>13</sup> FTU: Fictional Time Units. Nombre que se le ha dado a las unidades de tiempo utilizadas durante la generación y ejecución de los casos de prueba.

Conjunto de probabilidades	FIFO	LESS WORKLOAD
1	177.64	131.775
2	207.51	161.87

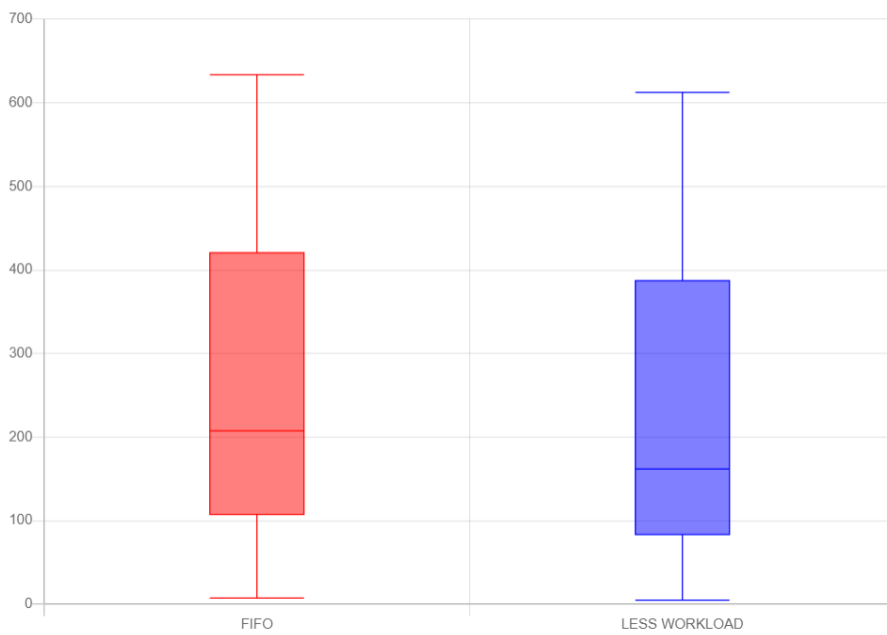
**Tabla 7:** Tiempo promedio de finalización de orden para ambos *schedulers* , expresados en FTU

Conjunto de probabilidades	FIFO (MAX - MIN)	LESS WORKLOAD (MAX - MIN)
1	622.82 - 9.027	585.87 - 7.126
2	633.69 - 7.565	612.64 - 5.112

**Tabla 8:** valores máximos y mínimos de tiempo de finalización de órdenes alcanzados durante las pruebas para ambos *schedulers* , expresados en FTU



**Figura 17:** tiempos (mínimo, promedio y máximo) de finalización de órdenes (conjunto de probabilidades 1)  
Fuente: elaboración propia.



**Figura 18:** Tiempos (mínimo, promedio y máximo) de finalización de órdenes (conjunto de probabilidades 2)  
Fuente: elaboración propia.

### 6.3 Análisis y comparaciones

Como puede apreciarse en los gráficos de las Figuras 17 y 18 y las Tablas 7 y 8, como era de esperarse, el *scheduler* LESS\_WORKLOAD presenta unos tiempos significativamente mejores a los tiempos logrados con el *scheduler* simple FIFO. Recordar de secciones anteriores que el *scheduler* LESS\_WORKLOAD tiene en cuenta no solo la distancia entre las ubicaciones geográficas de la tarea y las máquinas, sino también la carga de trabajo de cada máquina.

En las primeras 30 ejecuciones -que utilizaron las probabilidades 1-, el *scheduler* LESS\_WORKLOAD logró un tiempo promedio de finalización de órdenes menor al obtenido con el algoritmo FIFO, equivalente a un 25.82% de reducción de tiempo. Por otro lado, en las siguientes 30 ejecuciones -con el conjunto de probabilidades 2-, esta diferencia obtenida fue del 21.99%.

Con las pruebas realizadas se concluye que, aún con variaciones en las probabilidades que pueden tener los sucesos en entornos de fabricación en la nube, un algoritmo de planificación inteligente puede reducir significativamente el ETA<sup>14</sup> de las órdenes.

En el caso particular de los dos algoritmos implementados durante este trabajo, esto se debe principalmente a cómo funciona el algoritmo LESS\_WORKLOAD. Este algoritmo prioriza el envío de tareas a aquellas máquinas compatibles que tienen una menor carga de trabajo por delante, por lo que pueden comenzar -y, finalizar- la tarea más tempranamente.

Asimismo, cabe aclarar que, si bien a simple vista la diferencia puede no parecer tan significativa, como fue oportunamente mencionado en la sección anterior, estos valores se encuentran expresados en unidades ficticias de tiempo definidas por el equipo. Estas unidades ficticias surgieron con el objetivo de poder simular ciclos completos de trabajo en cuestión de minutos. Basta fijarse en el porcentaje de reducción de tiempo para notar que, cuando estos valores se extrapolan a tiempo real, la diferencia se torna aún más evidente.

---

<sup>14</sup> ETA: estimated time of arrival

## 7. Conclusiones y trabajos a futuro

El trabajo realizado puede dividirse en dos etapas. La primera de ellas, consistió en el estudio del concepto y estado del arte de CMfg, mientras que, para la segunda etapa, se puso en práctica gran parte de la información adquirida en la etapa anterior, desarrollando un prototipo para realizar una prueba de concepto de CMfg.

Durante la primera etapa, se realizó una revisión sistemática de la literatura de CMfg. En esta revisión se analizó el estado actual de CMfg, así como también las distintas áreas y problemas de estudio de este nuevo paradigma. En esta etapa del trabajo, pudo verificarse que el concepto de CMfg e Industria 4.0 ha estado tomando mucha fuerza en los últimos años y, a medida que el tiempo transcurre, es cada vez mayor el número de investigadores que realizan publicaciones al respecto cada año.

Para la segunda etapa de este trabajo, se estudió una nueva plataforma desarrollada por un equipo de investigadores alemanes llamada Eclipse BaSyx. Durante esta investigación se evaluaron los SDKs disponibles para distintos lenguajes de programación, así como también las ventajas y desventajas de hacer uso del kit de herramientas provisto por este. Una vez validada la herramienta mediante el desarrollo de un primer prototipo de software, se procedió a implementar un prototipo más complejo que permitiera simular las principales funciones de un sistema real de CMfg.

La prueba de concepto implementada, en conjunto con un proceso de experimentación de un caso de estudio, permitió vincular el uso de Eclipse BaSyx con una arquitectura tipo de CMfg. El caso de estudio permitió también mostrar algunas de las funciones esenciales de CMfg, como la reasignación de tareas entre diferentes máquinas ante eventuales fallas de forma de cumplir con el pedido de un cliente y la posibilidad de realizar mantenimientos preventivos sin afectar el despacho de tareas.

Finalmente, este diseño experimental, demostró la gran capacidad adaptativa que tiene CMfg en conjunto con BaSyx, donde es posible modificar en tiempo real la estrategia de asignación de recursos, monitorear el sistema, obtener información en tiempo real y realizar ajustes de acuerdo a necesidades puntuales que puedan aparecer en el sistema.

Para finalizar, de este trabajo se desprenden varias posibles líneas de trabajo a futuro, que incluyen desde mejoras a la implementación actual, profundizaciones de los temas investigados y nuevas vertientes de investigación.

Con respecto a posibles nuevas investigaciones, destaca la vertical de la resiliencia, entendiéndose por resiliencia la capacidad de un sistema para recuperar su estado inicial cuando cesa la perturbación a la que fue sometido<sup>15</sup>. Resulta de especial interés el estudio de técnicas que permitan fortalecer la resiliencia de los sistemas de Industria 4.0 para que las perturbaciones que puedan sufrir desvíen el sistema lo menos posible de su estado normal y que este regrese a él velozmente.

---

<sup>15</sup> De la definición de “resiliencia” según el Diccionario de la lengua española, de la Real Academia Española.

Una segunda línea de investigación que se desprende naturalmente de este trabajo es la distribución de los costos y beneficios del uso de sistemas de *Cloud Manufacturing* entre todos los agentes que lo componen. Como fue adelantado en la subsección de temas a resolver (Sección 2.4.5), dentro de la sección de estado del arte, la adopción de CMfg requiere un nuevo modelo de negocio que apoye la naturaleza colaborativa de este paradigma. Parte fundamental de este modelo de negocio se basa en la distribución de valor entre los agentes, que actúan como socios. Resulta entonces interesante abrir una línea de investigación sobre cómo realizar la distribución del trabajo -y en consecuencia, de los beneficios- entre las partes para que *Cloud Manufacturing* resulte un paradigma atractivo.

Estos dos primeros trabajos a futuro se relacionan con temas importantes que fueron tratados durante todo este trabajo, tales como la flexibilidad de los sistemas de producción -y de la producción misma- y la interoperabilidad entre diferentes máquinas, empresas y sistemas.

Como tercera línea de trabajo a futuro, resultaría interesante extender la implementación desarrollada, haciendo uso de máquinas físicas conectadas con BaSyx de forma de lograr un sistema completo de Cloud Manufacturing. En este sentido, también queda pendiente -puesto que se encontraba fuera del alcance de este proyecto- la implementación de nuevas características en el sistema actual, tales como: cálculos de costos de transporte, impuestos y demás a la hora de calcular a qué máquina será asignada una tarea y la discriminación entre proveedores a la hora de utilizar o no una máquina para asignarle trabajo (por ejemplo, para asegurar una distribución de trabajo justa para todos los proveedores).

Finalmente, un cuarto trabajo a futuro que se desprende del presente trabajo, consiste en la investigación de cómo vincular la Industria 4.0 con el Desarrollo Sostenible, que forma parte de la Agenda 2030 de la Asamblea General de la ONU<sup>16</sup>, que incluye entre sus objetivos la producción y consumo responsables, la innovación en industrias e infraestructura, el uso de energía no contaminante y demás acciones ecológicas.

---

<sup>16</sup> Fuente: Naciones Unidas  
(<https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible>)

## Glosario

Esta sección presenta algunos conceptos que son relevantes en toda la extensión del documento.

**Asset:** cualquier activo físico, proceso, persona, lugar, sistema o dispositivo.

**Algoritmo:** conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. [R.A.E.]

**Big data:** conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales. [Libro “Nuevas Tecnologías y Globalización”, Pavel Sidorenko Bautista, Editorial Elearning, ISBN: 978-84-17814-54-0]

**Digital Twin (DT):** réplicas digitales de activos físicos potenciales y reales, procesos, personas, lugares, sistemas y dispositivos que pueden utilizarse para diversos fines, todos los activos de las empresas, desde maquinarias hasta servicios.

**CMfg:** Cloud Manufacturing

**Cloud:** la nube, o Cloud, es una metáfora que se utiliza para designar una red mundial de servidores remotos, cada uno con una función única, que funciona como un único ecosistema, normalmente asociada a Internet. Estos servidores están diseñados para almacenar y administrar datos, ejecutar aplicaciones o entregar contenido o servicios.

**Cloud Computing (CC):** paradigma que permite ofrecer servicios de computación a través de una red, la cual suele ser internet. [Wikipedia]

**Everything as a Service (XaaS):** consiste en una gran variedad de servicios y productos que pueden ser accedidos por los usuarios bajo demanda a través de internet.

**Gemelos digitales:** ver definición en *Digital Twins*.

**Holismo; Holístico, ca:** posición metodológica y epistemológica que postula cómo los sistemas (físicos, biológicos, sociales, económicos, mentales, lingüísticos, etc.) y sus propiedades deben ser analizados en su conjunto y no sólo a través de las partes que los componen. [Wikipedia]

**Internet of Things (IoT):** concepto que se ha puesto de moda en los últimos años, consiste en un sistema de dispositivos interrelacionados con la capacidad de comunicarse entre ellos sin intervención humana. Es, en definitiva, la conexión de internet más con objetos que con personas. [Wikipedia]

**Matching:** Vinculación o emparejamiento entre dos o más elementos. En el contexto de este trabajo, por ejemplo, entre actores/usuarios del sistema, o entre un requerimientos y recursos de fabricación.



**Middleware:** software que proporciona un enlace entre aplicaciones de software independientes. Se utiliza para simplificar el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones entre sistemas distribuidos, redundando en una mejora de la calidad de servicio, la seguridad, etcétera. [Wikipedia]

**Nube:** ver definición en **Cloud**.

**Paradigma:** teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento. [R.A.E.]

**Pool:** se le llama *pool* virtual de servicios a un grupo o conjunto de servicios que se encuentran disponibles dentro de un sistema.

**Pyme:** acrónimo de pequeña y mediana empresa. Empresa mercantil, industrial, etc., compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación. [R.A.E.]

**Sistema ciber-físico:** mecanismo (sistema físico) controlado o monitorizado por algoritmos basados en computación y estrechamente integrados con Internet. [Wikipedia]

**Software:** conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. [R.A.E.]

## Bibliografía

- [1] Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, 57(15-16), 4854-4879.
- [2] Adamson, G., Wang, L., Holm, M., & Moore, P. (2017). Cloud manufacturing—a critical review of recent development and future trends. *International Journal of Computer Integrated Manufacturing*, 30(4-5), 347-380.
- [3] Wan, J., Tang, S., Hua, Q., Li, D., Liu, C., & Lloret, J. (2017). Context-aware cloud robotics for material handling in cognitive industrial Internet of Things. *IEEE Internet of Things Journal*, 5(4), 2272-2281.
- [4] Dietz, M., & Pernul, G. (2020). Digital twin: Empowering enterprises towards a system-of-systems approach. *Business & Information Systems Engineering*, 62(2), 179-184.
- [5] Wang, W., Fan, L., Huang, P., & Li, H. (2019). A New Data Processing Architecture for Multi-Scenario Applications in Aviation Manufacturing. *Ieee Access*, 7, 83637-83650.
- [6] Kim, D. B. (2019). An approach for composing predictive models from disparate knowledge sources in smart manufacturing environments. *Journal of Intelligent Manufacturing*, 30(4), 1999-2012.
- [7] Henzel, R., & Herzwurm, G. (2018). Cloud Manufacturing: A state-of-the-art survey of current issues. *Procedia CIRP*, 72, 947-952.
- [8] Moustafa, N., Adi, E., Turnbull, B., & Hu, J. (2018). A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access*, 6, 32910-32924.
- [9] Ren, L., Zhang, L., Tao, F., Zhao, C., Chai, X., & Zhao, X. (2015). Cloud manufacturing: from concept to practice. *Enterprise Information Systems*, 9(2), 186-209.
- [10] Babiceanu, R. F., & Seker, R. (2019). Cyber resilience protection for industrial internet of things: A software-defined networking approach. *Computers in Industry*, 104, 47-58.
- [11] Liu, C., Zhou, Y., Cen, Y., & Lin, D. (2019). Integrated application in intelligent production and logistics management: technical architectures concepts and business model analyses for the customised facial masks manufacturing. *International Journal of Computer Integrated Manufacturing*, 32(4-5), 522-532.
- [12] Shafiq, S. I., Sanin, C., Szczerbicki, E., & Toro, C. (2014). Implementing virtual engineering objects (VEO) with the set of experience knowledge structure (SOEKS). *Procedia Computer Science*, 35, 644-652.

- [13] Shafiq, S. I., Sanin, C., Szczerbicki, E., & Toro, C. (2015). Virtual engineering object/virtual engineering process: a specialized form of cyber physical system for Industrie 4.0. *Procedia Computer Science*, 60, 1146-1155.
- [14] Shafiq, S. I., Sanin, C., Toro, C., & Szczerbicki, E. (2016). Virtual engineering process (VEP): a knowledge representation approach for building bio-inspired distributed manufacturing DNA. *International Journal of Production Research*, 54(23), 7129-7142.
- [15] Yan, H., Wan, J., Zhang, C., Tang, S., Hua, Q., & Wang, Z. (2018). Industrial big data analytics for prediction of remaining useful life based on deep learning. *IEEE Access*, 6, 17190-17197.
- [16] Yu, S., & Xu, X. (2015, September). Development of a product configuration system for cloud manufacturing. In *IFIP International Conference on Advances in Production Management Systems* (pp. 436-443). Springer, Cham.
- [17] Mourtzis, D., Milas, N., & Vlachou, A. (2018). An internet of things-based monitoring system for shop-floor control. *Journal of Computing and Information Science in Engineering*, 18(2).
- [18] Caggiano, A. (2018). Cloud-based manufacturing process monitoring for smart diagnosis services. *International Journal of Computer Integrated Manufacturing*, 31(7), 612-623.
- [19] Kuru, K., & Yetgin, H. (2019). Transformation to advanced mechatronics systems within new industrial revolution: A novel framework in automation of everything (AoE). *IEEE Access*, 7, 41395-41415.
- [20] Liu, Y., Tong, K., Mao, F., & Yang, J. (2019). Research on digital production technology for traditional manufacturing enterprises based on industrial Internet of Things in 5G era. *The International Journal of Advanced Manufacturing Technology*, 1-14.
- [21] Ooi, B. Y., Kong, Z. W., Lee, W. K., Liew, S. Y., & Shirmohammadi, S. (2019). A collaborative IoT-gateway architecture for reliable and cost effective measurements. *IEEE Instrumentation & Measurement Magazine*, 22(6), 11-17.
- [22] Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4), 2039-2047.
- [23] Villalonga, A., Beruvides, G., Castano, F., & Haber, R. E. (2020). Cloud-based industrial cyber-physical system for data-driven reasoning: A review and use case on an industry 4.0 pilot line. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 16(9), 5975-5984.
- [24] Kabugo, J. C., Jämsä-Jounela, S. L., Schiemann, R., & Binder, C. (2020). Industry 4.0 based process data analytics platform: A waste-to-energy plant case study. *International Journal of Electrical Power & Energy Systems*, 115, 105508.

- [25] Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2019). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics*, 16(1), 183-192.
- [26] Simeone, A., Caggiano, A., Deng, B., Zeng, Y., & Boun, L. (2018). Resource efficiency optimization engine in smart production networks via intelligent cloud manufacturing platforms. *Procedia Cirp*, 78, 19-24.
- [27] Giraldo-Castrillon, F. A., Páramo-Bermúdez, G. J., & Muñoz-Betancur, J. M. (2019). Monitoring of Machining in the Cloud as a Cost Management Service and Follow of Cutting Parameters: Environment Developed With IoT Tools. *International Journal of Manufacturing, Materials, and Mechanical Engineering (IJMMME)*, 9(3), 26-41.
- [28] Simeone, A., Caggiano, A., Deng, B., & Boun, L. (2019). A deep learning based-decision support tool for solution recommendation in cloud manufacturing platforms. *Procedia CIRP*, 86, 68-73.
- [29] Fisher, O., Watson, N., Porcu, L., Bacon, D., Rigley, M., & Gomes, R. L. (2018). Cloud manufacturing as a sustainable process manufacturing route. *Journal of manufacturing systems*, 47, 53-68.
- [30] Banica, L., & Stefan, C. (2019). Stepping into the Industry 4.0: The Digital Twin Approach. *Annals of the University Dunarea de Jos of Galati: Fascicle: I, Economics & Applied Informatics*, 25(3).
- [31] Lim, K. Y. H., Zheng, P., & Chen, C. H. (2019). A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing*, 1-25.
- [32] He, B., & Bai, K. J. (2020). Digital twin-based sustainable intelligent manufacturing: a review. *Advances in Manufacturing*, 1-21.
- [33] Redelinghuys, A. J. H., Basson, A. H., & Kruger, K. (2019). A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing*, 1-20.
- [34] Zhong, R. Y., Xu, X., & Wang, L. (2017). IoT-enabled smart factory visibility and traceability using laser-scanners. *Procedia Manufacturing*, 10, 1-14.
- [35] Tong, X., Liu, Q., Pi, S., & Xiao, Y. (2019). Real-time machining data application and service based on IMT digital twin. *Journal of Intelligent Manufacturing*, 1-20.
- [36] Lee, J., Azamfar, M., Singh, J., & Siahpour, S. (2020). Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing. *IET Collaborative Intelligent Manufacturing*, 2(1), 34-36.

- [37] Simeone, A., Deng, B., & Caggiano, A. (2020). Resource efficiency enhancement in sheet metal cutting industrial networks through cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, 1-21.
- [38] Caggiano, A., Segreto, T., & Teti, R. (2020). Cloud manufacturing architecture for part quality assessment. *Cogent Engineering*, 7(1), 1715524.
- [39] Liu, Y. K., Zhang, X. S., Zhang, L., Tao, F., & Wang, L. H. (2019). A multi-agent architecture for scheduling in platform-based smart manufacturing systems. *Frontiers of Information Technology & Electronic Engineering*, 20(11), 1465-1492.
- [40] Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of manufacturing systems*, 47, 179-198.
- [41] Li, X., Wan, J., Dai, H. N., Imran, M., Xia, M., & Celesti, A. (2019). A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing. *IEEE Transactions on Industrial Informatics*, 15(7), 4225-4234.
- [42] Tang, H., Li, D., Wan, J., Imran, M., & Shoaib, M. (2019). A reconfigurable method for intelligent manufacturing based on industrial cloud and edge intelligence. *IEEE Internet of Things Journal*, 7(5), 4248-4259.
- [43] Grassi, A., Guizzi, G., Santillo, L. C., & Vespoli, S. (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*.
- [44] Villalobos, K., Ramírez-Durán, V. J., Diez, B., Blanco, J. M., Goñi, A., & Illarramendi, A. (2020). A three level hierarchical architecture for an efficient storage of industry 4.0 data. *Computers in Industry*, 121, 103257.
- [45] Lee, W. K., Phan, R. C. W., Poh, G. S., & Goi, B. M. (2018). SearchaStore: fast and secure searchable cloud services. *Cluster Computing*, 21(2), 1189-1202.
- [46] BA, P., PN, S., & PM, S. (2020). Shop floor to cloud connect for live monitoring the production data of CNC machines. *International Journal of Computer Integrated Manufacturing*, 33(2), 142-158.
- [47] Canizo, M., Conde, A., Charramendieta, S., Minon, R., Cid-Fuentes, R. G., & Onieva, E. (2019). Implementation of a large-scale platform for cyber-physical system real-time monitoring. *IEEE Access*, 7, 52455-52466.
- [48] Mullany, M. (2018, April 18). Percentage of Customers That Adopt An Innovation Over Time [Illustration]. *The Power of the Adoption Curve*.  
<https://www.linkedin.com/pulse/basics-adoption-curve-michael-mullany/>

[49] Kuhn, T. (2019, June 19). AAS and submodels structure information in a tree of properties [Illustration]. BaSyx Documentation.  
<https://wiki.eclipse.org/File:BaSyx.AASFormat.png>

[50] Kuhn, T. (2019, June 19). Asset Administration Shell based BaSyx architecture example [Illustration]. BaSyx Documentation.  
<https://wiki.eclipse.org/File:BaSyx.AAS.ArchitectureExample2.png>

[51] Kuhn, T. (2018, November 29). Virtual Automation Bus example [Illustration]. BaSyx Documentation. <https://wiki.eclipse.org/File:Basyx.VABExample.png>

[52] Kuhn, T. (2018, August 28). Main BaSyx Components [Illustration]. BaSyx Documentation. [https://wiki.eclipse.org/File:BaSyx.BaSyx10Mins\\_8.png](https://wiki.eclipse.org/File:BaSyx.BaSyx10Mins_8.png)

[53] Kuhn, T. (2020, November 13). Architecture overview and the most important components of eclipse BaSyx [Illustration]. BaSyx Documentation.  
<https://wiki.eclipse.org/File:Basyx.overview.png>

# Anexos

# Anexo I

## Revisión de la literatura



Estado del arte

# Arquitecturas y plataformas de Cloud Manufacturing

## Resumen

La necesidad de transformar los elementos de la naturaleza para poder hacer uso de ellos, ha estado presente desde el origen del ser humano. Durante siglos, hemos realizado diferentes actividades que han permitido satisfacer distintos tipos de necesidades. Actividades que, si bien han variado sustancialmente a lo largo de la historia para adaptarse, por ejemplo, a nuevas tecnologías, siempre mantienen una misma columna vertebral.

Se le llama “actividad económica” al procedimiento mediante el cual se producen e intercambian bienes y servicios con el fin de satisfacer las necesidades de los consumidores, consiguiendo una renta por ello. En este sentido, la industria, es aquella actividad económica que emplea y transforma materias primas en productos elaborados o semielaborados, haciendo uso de alguna fuente de energía, maquinaria y recursos humanos.

Los principales problemas que han acompañado el desarrollo de la actividad industrial durante toda la historia incluyen -mas no se limitan a- la reducción de costos de producción, el aumento de la cantidad de bienes y servicios producidos, la optimización del uso de recursos, la reducción del tiempo necesario para transformar un recurso en un producto útil y la expansión de los mercados. Todo esto ha generado una serie de “revoluciones” que, de alguna forma, ofrecen soluciones a estos problemas.

Cloud Manufacturing (o fabricación en la nube) forma parte de la llamada Cuarta Revolución Industrial.

Este trabajo generará un marco teórico actualizado realizando un relevamiento sobre el estado del arte de Cloud Manufacturing, buscando recopilar de diferentes fuentes las características de este nuevo paradigma de fabricación y los avances que se han suscitado, así como presentar los desafíos que aún resta por resolver y hacia dónde deberían dirigirse los trabajos a futuro.

## Palabras claves

Cloud Manufacturing (CMfg), Cloud Computing (CC), Cyber-Physical Production Systems (CPPS), Everything as a Service (XaaS), Big Data, Robotics, Industry 4.0, Internet of Things (IoT), Smart Factory, Cloud Platform, Enterprise Information Systems

## Tabla de contenido

1.	Introducción	1
1.1.	Contexto e historia	1
2.	Investigación	3
2.1.	Motivación	4
2.2.	Beneficios de Cloud Manufacturing	5
2.3.	Características	6
2.4.	Puntos claves de un sistema de CMfg	15
2.5.	Ejemplos de uso	17
2.6.	Temas a resolver	18
2.7.	Investigación de arquitecturas y plataformas de CMfg	20
2.8.	Resultados y discusión	23
3.	Conclusiones	38
	Glosario	39
	Referencias	41

# 1. Introducción

El objetivo de este estudio es realizar una investigación acerca del estado del arte de Cloud Manufacturing y de la Industria 4.0 en general, explicando sus características, los avances que se han tenido y las problemáticas que se enfrentan al trabajar con un paradigma que se encuentra en pleno desarrollo.

Para comprender el problema es necesario, primero, conocer el contexto histórico que posibilitó al ser humano alcanzar la presente Cuarta Revolución Industrial.

## 1.1 Contexto e historia

El primer gran proceso de transformación económica, social y tecnológica del que se tiene registro en la época moderna, es la llamada Primera Revolución Industrial<sup>17</sup> (a veces también llamada simplemente Revolución Industrial), que tuvo lugar en la segunda mitad del siglo XVIII en Gran Bretaña, donde se vivió el mayor conjunto de transformaciones económicas, tecnológicas y sociales de la historia de la humanidad desde el Neolítico.

En un principio, la economía era básicamente rural, basada fundamentalmente en la agricultura y el comercio. Esta Primera Revolución Industrial, que se extendió aproximadamente desde el año 1760 hasta el año 1840, introdujo la máquina de vapor como gran mejora tecnológica, que permitía transformar la energía térmica contenida en el vapor de agua en energía mecánica. La economía, entonces, pasó a tener un carácter urbano, industrializada y mecanizada.

Aproximadamente treinta años más tarde, en 1870, comenzó a producirse lo que se conoce como la Segunda Revolución Industrial<sup>18</sup>, también llamada revolución del transporte, cuya característica principal fue la globalización, la internacionalización de la economía. Este nuevo paradigma de fabricación introdujo los motores de combustión interna, la producción en masa, la línea de montaje y el uso de la electricidad como principales innovaciones para la industria. El período de esta revolución se extendió aproximadamente hasta el año 1914.

---

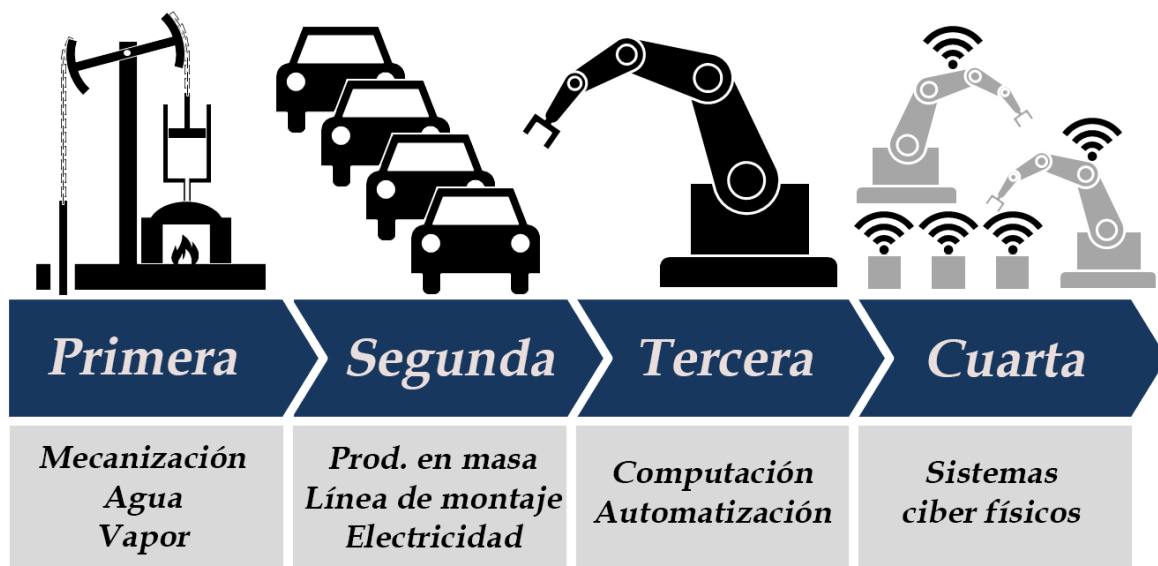
<sup>17</sup> Fuentes: "Industrial Revolution", History (<https://www.history.com/topics/industrial-revolution/industrial-revolution>)  
"The 4 Industrial Revolutions", Institute of Entrepreneurship Development (<https://www.history.com/topics/industrial-revolution/industrial-revolution>)

<sup>18</sup> Fuentes: "The Second Industrial Revolution has Brought Modern Social and Economic Developments", Haradhan Kumar Mohajan, ISSN: 2381-7771

La Tercera Revolución Industrial<sup>19</sup> es la más cercana en el tiempo y es la que sentó las bases de la economía y la industria actual. Conocida también como Revolución Científico-Tecnológica, modificó el paradigma de fabricación de la época, introduciendo la utilización de la informática y la automatización, donde la robótica pasa a ser parte fundamental del proceso de producción. Sumado a estas características, también destacan la utilización de energías renovables, las tecnologías de almacenamiento de energía y el desarrollo de la red eléctrica inteligente y de los vehículos eléctricos.

La economía se encuentra ahora en el umbral de la cuarta revolución industrial, etapa que ha comenzado recientemente y cuyo desarrollo está proyectado hacia la tercera década del siglo XXI, teniendo como principales características la utilización de sistemas ciber físicos, inteligencia artificial, utilización de big data e interconexión masiva de sistemas y dispositivos digitales, formando así las llamadas Fábricas Inteligentes (Smart Factories)

La Figura 1 muestra, en forma de diagrama, las características principales que destacan y definen a las primeras cuatro revoluciones industriales que la humanidad ha conocido.



**Figura 1:** Diagrama de las primeras cuatro revoluciones industriales.

Fuente: “A Critical Look at Industry 4.0”, by Christoph Roser at AllAboutLean.com under the free CC-BY-SA 4.0 license.

<sup>19</sup> Fuentes: Roel, V. (1998). La tercera revolución industrial y la era del conocimiento. Lima, Universidad Editorial Mayor de San Marcos, Fondo Editorial.  
Cotón, S. H., & Gutiérrez, J. S. (2016). Las consecuencias de la tercera revolución industrial. *Mercados y Negocios* (2594-0163 en línea; 1665-7039 en impreso), (8), 11-20.

## 2. Investigación

Cloud Manufacturing o Fabricación en la nube, es el nuevo paradigma que posibilita la implementación de la llamada Industria 4.0<sup>20</sup>.

Gran parte de la literatura, 37%, según describen Henzel et al [7], se enfoca en el concepto de CMfg. Esto indica que la investigación, en general, se encuentra en una etapa muy temprana. Por otra parte, el 35% de la literatura, según los mismos autores, trata sobre revisiones de otras literaturas

La primera literatura sobre CMfg de la que se tiene registro, data del año 2010 y la cantidad de publicaciones ha aumentado año tras año. El incremento de literatura en el año 2018 con respecto al incremento del 2017, por ejemplo, fue de un 40% más (Ellwein et al. [5]); esto muestra que es un paradigma completamente nuevo y que se encuentra situado justo en la frontera del conocimiento.

Tomando las tres bases de datos que cuentan con mayor cantidad de literatura sobre el tema, las proporciones se dividen, según muestran Ellwein et al en [5], en Springer (43.81%), Elsevier (39.05%), IEEE (17.14%).

Resulta de sumo interés, también, conocer cómo es la distribución de la literatura entre diferentes países a lo largo del mundo. La investigación desarrollada por el equipo de Ellwein muestra que los países que más literatura han publicado acerca del tema son (en orden) China -país que ha aportado, por lejos, mayor cantidad de literatura- con un 37% de la misma, seguido por Estados Unidos, quien ha contribuido con un 8% de los artículos y Alemania, con una literatura que representa el 6% del total. Un poco más alejados en cuanto a porcentaje, se han realizado varios aportes literarios desde Reino Unido, Suecia, Taiwán, Italia, India, Nueva Zelanda y Grecia.

Para el correcto desarrollo de la investigación, se plantean en Tabla 1 los diferentes nombres que se encuentran fuertemente relacionados con Cloud Manufacturing.

Teniendo en cuenta que se trata de un paradigma nuevo que se encuentra aún en construcción, no resulta sorprendente el hecho de que se utilicen diferentes nombres para hacer mención de una misma idea o concepto.

---

<sup>20</sup> "Was ist Industrie 4.0?", Plattform Industrie 4.0, Bundesministerium für Wirtschaft und Energie (<https://www.plattform-i40.de/PI40/Navigation/DE/Industrie40/WasIndustrie40/was-ist-industrie-40.html>)

<b>Nombres relacionados con Cloud Manufacturing</b>
Factory of the future - FoF <sup>21</sup>
Industry 4.0 (Alemania)
Advanced Manufacturing (Estados Unidos)
e-Factory (Japón)
Intelligent Manufacturing (China)

**Tabla 1:** Nombres relacionados con Cloud Manufacturing  
Fuente: elaboración propia.

## 2.1 Motivación

El inicio de las revoluciones industriales, por lo general, sientan sus bases en alguna necesidad que es preciso resolver y en algún avance (sea tecnológico, de conocimiento, o de cualquier otra índole) que permita dicha resolución. En este sentido, parece de especial interés conocer el porqué de que esta nueva revolución industrial está teniendo lugar en esta época.

Con la economía altamente globalizada y el acceso masivo a Internet, las organizaciones se están enfrentando con nuevos retos y nuevas problemáticas que no pueden satisfacerse de manera óptima con el paradigma de fabricación actual. Por otra parte, además, existe actualmente un cúmulo de características, tecnologías y conocimientos que permiten pensar en la factibilidad de reinventar la forma de fabricación utilizada por las industrias.

Si bien el concepto de Internet de las cosas (Internet of Things, IoT) fue propuesto por Kevin Ashton en el año 1999<sup>22</sup>, su uso ha madurado en los últimos años, y está teniendo su “explosión” en la actualidad.

El Internet ya es un recurso accesible por todos y su uso se encuentra naturalizado e internalizado en todas las empresas. Esto, sumado al gran proceso de globalización que se ha generado, hace que las empresas puedan enfocarse en diseños de productos más complejos y en pasar de una fabricación orientada a productos a una fabricación orientada a servicios, donde ahora existe el concepto de “lote de tamaño 1”, que surge de la necesidad de satisfacer requerimientos cada vez más específicos de los consumidores.

Como contrapartida, existen actualmente tecnologías suficientes como para comenzar a llevar a cabo la modernización de los procesos de producción de las industrias para

<sup>21</sup> El término Factory of the future se utiliza para indicar la “nueva revolución industrial”, iniciada por una nueva generación de sistemas de fabricación concebidos para ser adaptativos, totalmente conectados, analíticos y altamente eficientes.

<sup>22</sup> Fuente: Kramp, T., Van Kranenburg, R., & Lange, S. (2013). Introduction to the Internet of Things. In *Enabling Things to Talk* (pp. 1-10). Springer, Berlin, Heidelberg.

adaptarse a estas características y cuyo objetivo, al igual que en el pasado, es maximizar la calidad, escalabilidad, flexibilidad, agilidad y adaptabilidad, minimizando también los costos.

## 2.2 Beneficios de Cloud Manufacturing

Para que un nuevo paradigma sea aceptado e incorporado, tiene que ofrecer algún beneficio importante para quien lo implemente. En el caso de Cloud Manufacturing son las industrias quienes implementan los paradigmas de fabricación.

La presente sección pretende describir de forma general quiénes y cómo se ven beneficiados por la implementación de CMfg.

La fabricación en la nube puede ser implementada tanto por grandes empresas como por aquellas de porte pequeño y mediano, y el uso de CMfg varía en función del tamaño de cada empresa. Con respecto a esto, parece ser una oportunidad única para las pymes, que son las que esencialmente se beneficiarían de este paradigma puesto que les permitiría actuar, virtualmente, como si fuesen empresas grandes. Se trata de utilizar la fabricación en la nube para implementar una fabricación cooperativa.

Por otra parte, al hacer uso de IoT, Cloud Manufacturing permite incluso interactuar desde la nube con los activos de producción. Al virtualizar los activos de las empresas, es posible monitorear el estado de los mismos, así como adelantarse a posibles fallas de funcionamiento y organizar los mantenimientos.

Esta última característica permite generar una gran base de conocimiento, obteniendo datos de una cantidad importante de empresas y dominios diferentes, lo cual permite realizar mejores análisis, correcciones y mejoras de los procedimientos, ofreciendo la posibilidad de aprender de esta información relevada para así mejorar continuamente el sistema y optimizar los flujos.

La idea principal sobre la que se rige Cloud Manufacturing es la de proveer como servicios todo tipo de recursos de fabricación (XaaS), vinculando de forma automática y eficiente usuarios con necesidades, con proveedores de recursos que puedan satisfacer dichas necesidades durante todo el proceso de fabricación.

Este enfoque permite escalar la producción muy fácilmente sin incurrir en grandes costos, ya que una empresa no tiene por qué comprar maquinaria nueva o más camiones o grandes almacenes, sino que puede “contratar mayores capacidades” en la nube. De la misma forma, al disminuir la producción, la empresa contrata menores capacidades en la nube y no se queda con maquinaria ociosa en su poder, lo que redundaría en gastos innecesarios. Por eso, una probable característica que debería ofrecer quien implemente un sistema de Cloud Manufacturing es la de “pago por uso”.

Finalmente, como el proceso de vinculación de proveedores y clientes es automático y cada cliente puede vincularse con un proveedor diferente para cada trabajo específico, se evitan los altos costos presentes en el paradigma de fabricación tradicional, asociado a los costos generados por los subcontratos entre empresas.



Por todo lo anteriormente mencionado, un paradigma de fabricación en la nube minimizaría muchos de los riesgos que actualmente aquejan a las empresas a lo largo de todo el ciclo de producción.

### 2.3 Características

Cloud Manufacturing (fabricación en la nube) es un nuevo paradigma de fabricación que incluye diseño, producción, fabricación, pruebas (testing), logística, mantenimiento y todas las demás etapas del ciclo de vida de un producto. Su objetivo es ofrecer servicios de fabricación “on-demand” a los consumidores, mediante Internet.

Lei Ren et al. [9], describen algunos de los conceptos utilizados en Cloud Manufacturing. Para una mejor comprensión del resto del documento y como base para discusiones posteriores, se resumen los conceptos fundamentales de CMfg en la Tabla 2 a continuación:

Concepto	Definición
Recurso de fabricación (Manufacturing Resource)	Entidad que puede soportar una actividad o función involucrada en el ciclo de vida de un producto. Los “recursos blandos” (soft resources) podrían ser el software, datos, información o conocimiento u otros elementos intelectuales, mientras que los “recursos duros” (hard resources) podrían ser las células de fabricación o el hardware de IT.
Capacidad de fabricación (Manufacturing Capability)	Competitividad profesional (recursos de fabricación, ventajas intelectuales, créditos) para emprender un trabajo en un ciclo de vida del producto, que representa la competencia (reducción de costos, mejora de calidad) que una organización o un individuo tiene para realizar una tarea específica y lograr un objetivo particular.
Plataforma de fabricación en la nube (Cloud Manufacturing Platform)	Entidad que administra un conjunto compartido de recursos y capacidades de fabricación a través de una red, que ofrece infraestructura y herramientas integradas basadas en IT para que proveedores y demandantes ofrezcan y consuman, respectivamente, “on demand” servicios en la nube.
Servicio de fabricación en la nube (Cloud Manufacturing Service)	Función basada en una capacidad de fabricación que puede lograr un objetivo en una actividad de un ciclo de vida del producto (servicio de diseño, de producción, de prueba, de gestión). Cuando es OnCloud, el control lo tiene una plataforma en la nube; cuando es OffCloud, necesita operaciones adicionales por parte de un operador de una plataforma en la nube.
Usuario de la nube (Cloud User)	Actor que participa en la fabricación en la nube. Hay tres tipos básicos, proveedor, consumidor y operador.
Virtualización de recursos	Proceso de mapeo de un recurso de fabricación real a uno

(Resource Virtualization)	lógico.
Capacidad de servitización (Capability Servitization)	Proceso de encapsulación desde una descripción abstracta de una capacidad de fabricación hasta un servicio estándar en la nube de acuerdo con una especificación
Sistema de fabricación en la nube (Cloud Manufacturing System)	Sistema que consiste en usuarios de la nube, plataforma en la nube, recursos y capacidades de fabricación, que admite aplicaciones específicas en un dominio de fabricación

**Tabla 2:** Conceptos utilizados en Cloud Manufacturing  
Fuente: elaboración propia.

## Actores

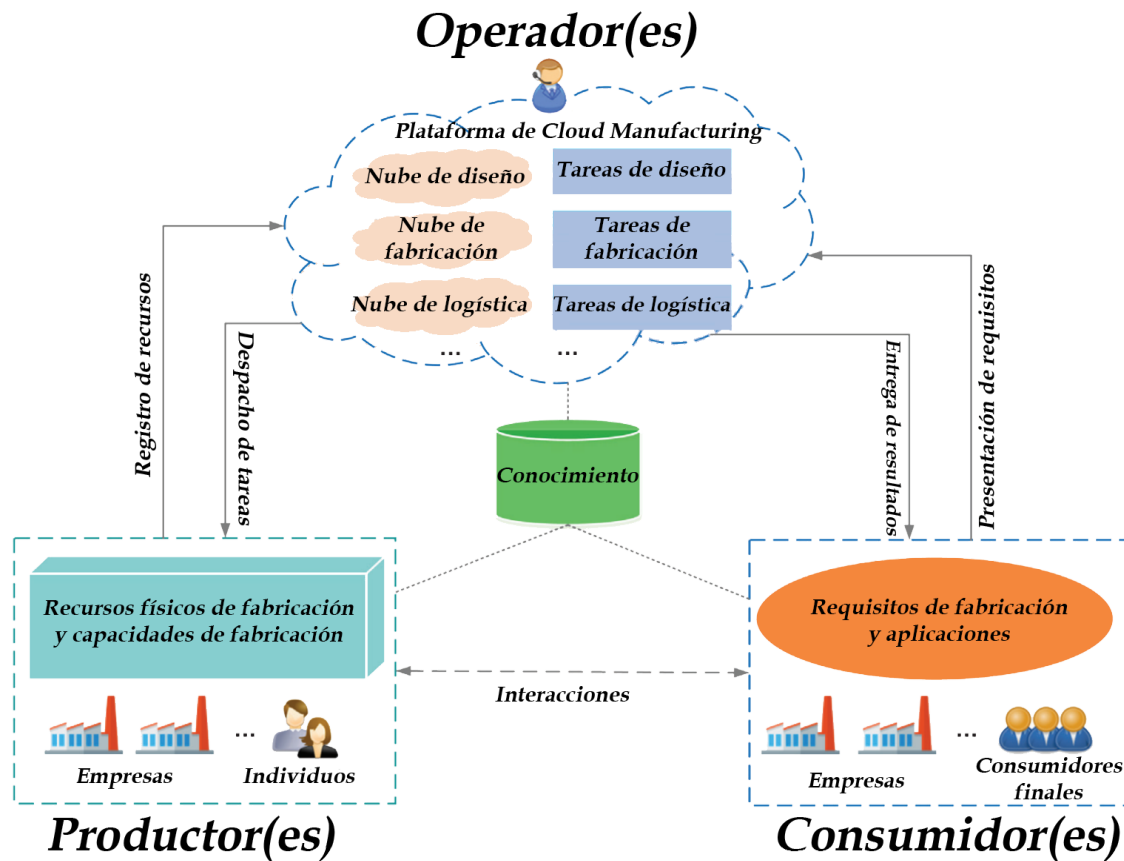
Independientemente de qué literatura se estudie, la investigación demuestra que toda concepción de alguna plataforma que implemente Cloud Manufacturing, estará formada por cuatro grandes partes bien definidas: tres actores (o stakeholders) y una base de conocimiento.

Los tres tipos de actores que participan de una plataforma de CMfg son los que se detallan junto al concepto “usuario de la nube” en Tabla 2, cada uno con funciones bien definidas.

A grandes rasgos,

1. **Los clientes (o consumidores)** son aquellos usuarios que necesitan obtener materia prima o servicios que no pueden -o no les rinde- producir, por ejemplo, porque la maquinaria es demasiado costosa. En esencia, tienen una necesidad que deben satisfacer.
2. **Los productores (o proveedores)** son aquellos usuarios que tienen los medios para satisfacer la necesidad de los clientes, por ejemplo, poseen maquinaria con capacidad ociosa.
3. **Los administradores de la nube, también llamados operadores** son aquellos usuarios encargado de conectar a los clientes con los productores de la manera más conveniente posible.
4. Finalmente, **la base de conocimiento** es un tipo especial de base de datos para la gestión del conocimiento, que le permite al sistema tomar decisiones de forma eficiente.

En [Figura 2] se muestra, esquematizado, el modelo de operación con estos tres actores y la base de conocimiento.



**Figura 2:** Modelo de operación de fabricación en la nube  
 Fuente: Liu et al. (2019) [1]

Las funciones que debe ofrecer una plataforma de CMfg, en cambio, no se encuentran tan estandarizadas como sí lo están los actores, sin embargo, hay una serie de funciones en las que parece haber consenso. Por hacer las veces de intermediario entre consumidores y productores, como por cumplir el rol de administrador del sistema, una plataforma de fabricación en la nube debe proporcionar funcionalidades básicas para ayudar a proveedores y clientes de la nube a alcanzar sus objetivos.

Ren et al. [9], definen algunas de estas funciones que se espera que provea un sistema de CMfg a este respecto.

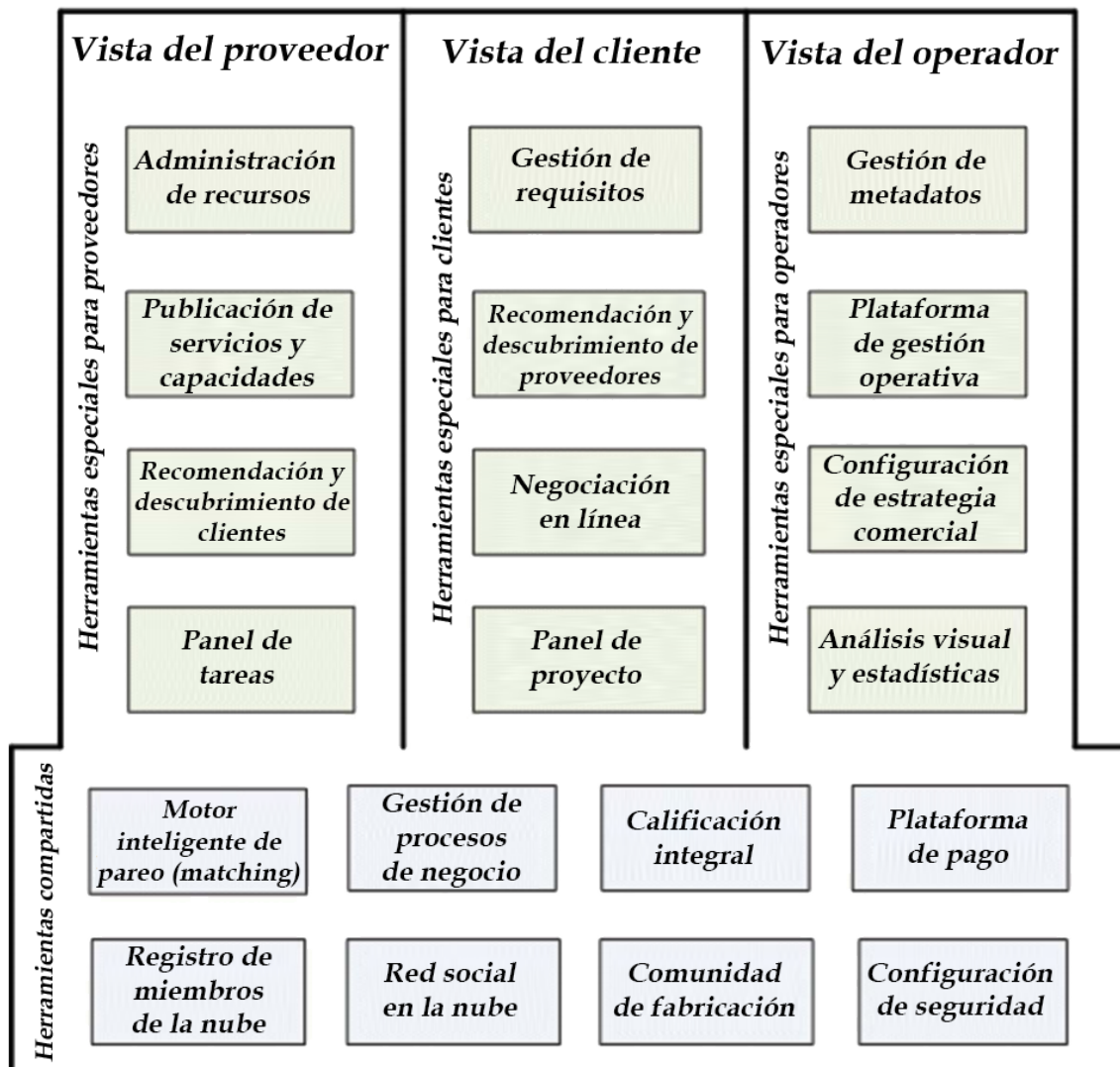
### **Funciones**

En primera instancia, por ser una plataforma de fabricación, se debería permitir la conexión de una variedad de recursos de fabricación, identificando, además, información clave de los recursos conectados. Más adelante, en este mismo documento, se explayará más sobre este punto, puesto que esta función sienta las bases para el resto de la operativa de la plataforma.

Relacionado con el punto anterior, una plataforma de CMfg debería también ser capaz de transformar recursos de fabricación reales en recursos virtuales. Como se desarrollará más adelante en esta investigación, una forma de virtualizar lógicamente un recurso físico es a través de los llamados “gemelos digitales” (o “*digital twins*”, en inglés).

Para una correcta y eficiente personalización flexible bajo demanda, estos recursos de fabricación virtualizados deberían ser agrupados para formar así conjuntos de recursos virtuales, que deberán, a su vez, estar correctamente administrados.

Una vez virtualizados y agrupados en conjuntos de recursos, la siguiente función que una plataforma de CMfg debería ofrecer, es la servitización de las capacidades de fabricación. Se le llama servitización a la transformación de las capacidades de fabricación en servicios en la nube que cumplen con una especificación estándar, añadiendo una descripción completa y efectiva de cada una de las capacidades específicas según cierta semántica.



**Figura 3:** Herramientas compartidas y especiales en la capa de kit de herramientas  
Fuente: Ren et al. (2015) [9]

Con respecto a los servicios, es preciso también contar con una forma de gestionarlos en la nube. Para ello, es necesario que el sistema provea la posibilidad de publicar servicios, realizar una vinculación (o *matching*) inteligente, componer diferentes servicios, gestionar el tiempo de ejecución y calificarlos. Análogamente a los recursos virtuales, los servicios en la

nube también deberán agruparse en conjuntos (o catálogos) de servicios, que construirán la base sobre la que operarán los sistemas de fabricación en la nube

El poderoso soporte de conocimiento y datos, por su parte, trae aparejada una función realmente importante que no debe faltar en una plataforma de CMfg, que consiste en una buena gestión del conocimiento y *big data*, que termina convirtiéndose en una parte esencial en la interoperabilidad y la colaboración empresarial. A medida que los recursos y las empresas que forman parte de la plataforma en la nube crecen, la gestión del llamado “big data” debe tomarse como una actividad seria.

Alguna de esta información que se genera, puede permitir, entre otras cosas, anticipar mantenimientos y fallas de los recursos físicos, ahorrando tiempo y dinero para las empresas, además de permitir tomar decisiones inteligentes dinámicamente ante las fallas que puedan ocurrir.

Otra funcionalidad que es también esperable de una plataforma de fabricación en la nube, consiste en el descubrimiento automático de servicios frente a las solicitudes enviadas por los clientes donde, una vez que productor y consumidor confirmen una solución, la plataforma debe configurar dinámicamente un sistema de fabricación virtual mediante el ensamblaje de servicios de colaboración, así como la integración de los recursos que sean relevantes.

El emparejamiento -o *matching*- entre cliente y proveedor puede ofrecerse de dos formas diferentes, llamadas “*push*” y “*pull*”, donde el emparejamiento de tipo “*push*” consiste en que la plataforma de CMfg periódicamente realiza un análisis proactivo semántico y multidimensional de la información de los proveedores y de los consumidores, y genera sugerencias o emparejamientos inteligentes para cada uno de ellos y los resultados se “*pushean*” a quienes más probablemente les serían de utilidad, es decir, en esencia, los clientes publicarían sus requerimientos y esperarían que el sistema les sugiera productores; mientras que el emparejamiento de tipo “*pull*” es realizado generalmente por los consumidores, quienes se encargan de buscar productores en un *pool* virtual de productores

En este sentido, es necesario también garantizar que el sistema de fabricación virtual se ejecute según lo esperado. Para ello, la plataforma en la nube debe monitorear el estado de tiempo de ejecución y responder a los cambios. Según los procesos de tareas de fabricación, será necesario coordinar diferentes proveedores de servicios entre los servicios, ya sean servicios *OnCloud* u *OffCloud*, así como proporcionar funciones de autodiagnóstico y autoreparación, además de autoadaptación ante excepciones que ocurran en tiempo real.

Desde una perspectiva comercial, las plataformas en la nube desempeñarán un papel de mercado comercial, donde los proveedores de la nube venden servicios y los clientes los compran. Es por esto que resulta necesario crear un entorno empresarial en el que tanto vendedores como compradores puedan manejar asuntos comerciales como negociaciones y contratos. Con respecto a esto, se espera que la nube pueda admitir también el esquema de pago llamado “pago por uso”, que consiste en capturar el consumo real de datos y realizar el cobro por el consumo realizado en lugar de tarifas fijas, haciendo que el costo en el que incurre una empresa al utilizar Cloud Manufacturing no resulte tan elevado.

Al desempeñar este papel comercial, y añadiendo el objetivo de construir una sociedad calificada de usuarios de fabricación en la nube, es necesario contar con un sistema de evaluación justa. Para ello, las plataformas deben admitir no solamente una calificación subjetiva por parte de las personas, sino también una calificación objetiva por parte de la plataforma, mediante el uso de datos de historial reales.

Esta calificación objetiva resulta de suma importancia para los clientes a la hora de seleccionar proveedores de servicios, ya que cuando un proveedor es más confiable, existe una mayor probabilidad de que sea elegido.

Una forma de promover una colaboración más profunda entre los usuarios de la nube, es construyendo una “red social” donde fluyan materiales, capital e información. Esta “red social” puede ser también una vía de coordinación entre los diferentes usuarios de la plataforma.

Finalmente, es necesario que las plataformas de CMfg cuenten también con una interfaz personalizada para cada tipo de usuario y cada rol, así como resulta importante también contar con una serie de políticas de seguridad que los usuarios finales puedan personalizar. Con respecto a esto último, hay consenso en que una correcta seguridad y buena privacidad es esencial para cualquier organización, no existen dos visiones al respecto. En cuanto a lo anterior, una interfaz de usuario individualizada para cada rol de usuario final -como ingenieros, diseñadores, trabajadores de línea de producción, verificadores, personal de marketing y gerencia- se hace necesaria como apoyo para su actividad específica. Una buena experiencia de usuario resulta importante en muchas formas y mejora la productividad y la visión que los usuarios tienen sobre el sistema.

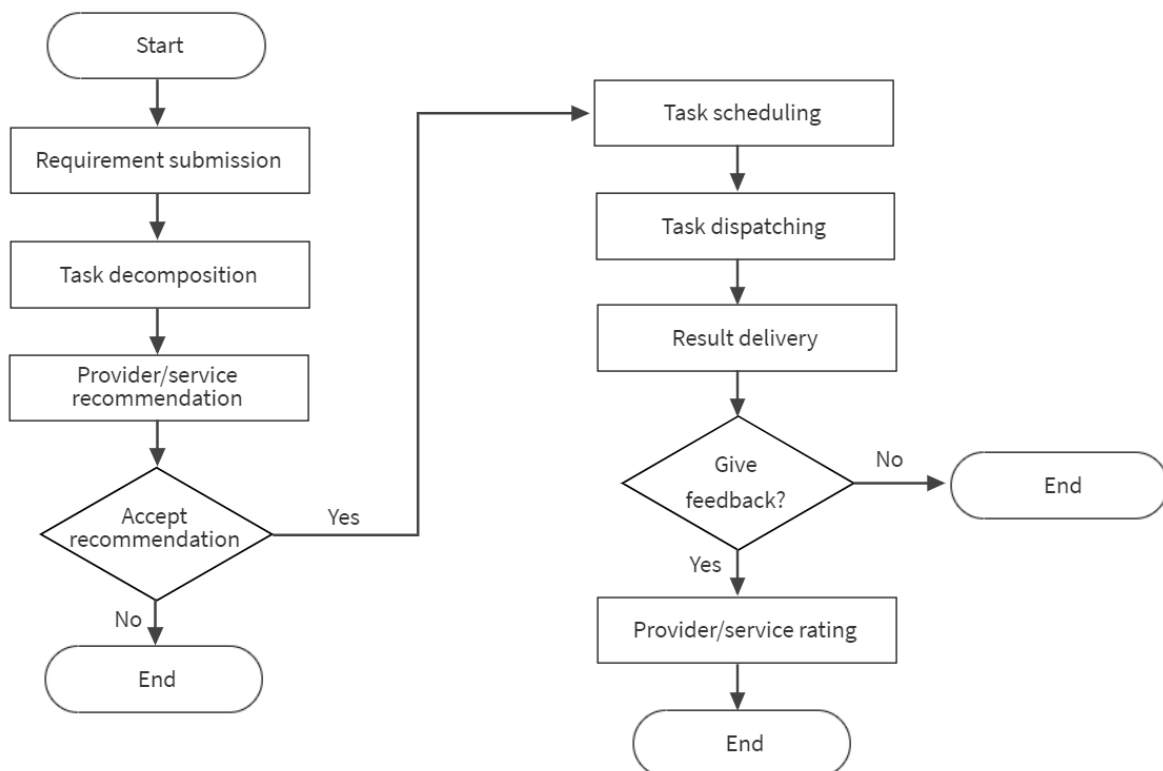
Con respecto al uso de Cloud Manufacturing, la idea es ofrecer dinamismo, permitiéndole a los productores ofrecer la capacidad de producción ociosa y poder variar la oferta en el tiempo. La nube debe responder automáticamente a estos cambios de capacidades para reacomodar eficientemente las asignaciones de recursos de los nuevos pedidos de los clientes que van llegando.

Como contrapartida, las plataformas de Cloud Manufacturing mejoran la competitividad de las empresas que lo implementan, puesto que buscan optimizar todo el proceso de producción.

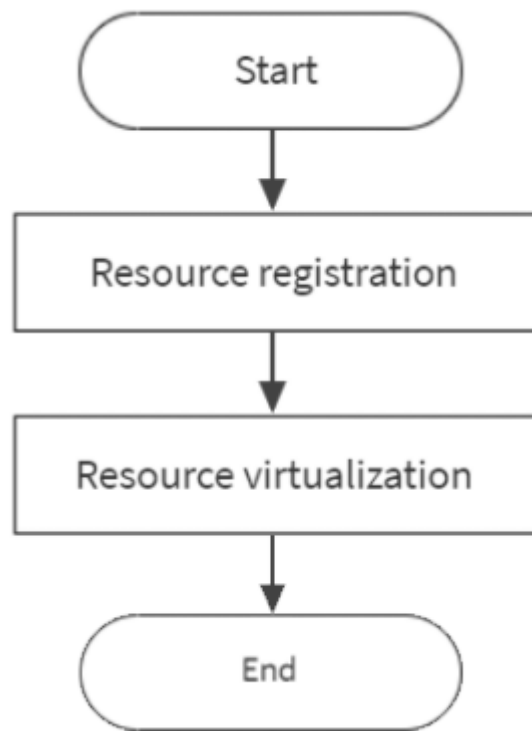
## Tareas que debe desarrollar un sistema de CMfg

Las figuras 4 y 5 muestran dos diagramas de flujo correspondientes a los casos de uso típicos de los dos tipos de usuario que conforman un sistema de Cloud Manufacturing, a excepción de los usuarios administradores de la nube.

La primera imagen corresponde al flujo del caso de uso de un cliente, quien presenta sus requisitos que deberán satisfacerse mediante servicios en la nube, mientras que la segunda, muestra el caso de uso de registro y virtualización de recursos por parte de un usuario proveedor, que serán utilizados para satisfacer las necesidades de los clientes.



**Figura 4:** Diagrama de flujo de tareas de un sistema de CMfg que formar parte de un caso de uso típico para el consumo de servicios por parte de un usuario de tipo cliente. Fuente: elaboración propia.



**Figura 5:** Diagrama de flujo de tareas de un sistema de CMfg que formar parte de un caso de uso típico para el registro y virtualización de recursos por parte de un usuario de tipo proveedor.

Fuente: elaboración propia.

En esencia, las tareas que debería ofrecer un sistema de Cloud Manufacturing consistirían en las que se listan a continuación:

1. **Registro de recursos (Resource registration):** Los fabricantes deben tener la posibilidad de ofrecer sus servicios/recursos a través de la plataforma.
2. **Virtualización de recursos (Resource virtualization):** Con el objetivo de mantener controlado el estado del sistema y de poder satisfacer las necesidades de los clientes, es preciso virtualizar los recursos que son ofrecidos por los proveedores. Existen desafíos interesantes en este sentido, particularmente porque los componentes físicos de los sistemas introducen diferencias cualitativas para con su representación orientada a objetos utilizada normalmente.
3. **Presentación de requisitos (Requirement submission):** Análogamente al registro de servicios y recursos que es requerido por los proveedores, los clientes necesitan poder especificar en la plataforma de Cloud Manufacturing todos los requisitos para llevar a cabo su producción.
4. **Descomposición de tareas (Task decomposition):** Una tarea, generalmente, puede descomponerse en subtareas con un grado de complejidad inferior. Es altamente útil poder descomponer en sub tareas todas aquellas tareas complejas que no puedan ser desarrolladas por un único servicio.



5. **Planificación (Scheduling):** La planificación es un proceso de organización, control y optimización del trabajo que resulta ser una tarea de vital importancia para CMfg. En el contexto de CMfg se refiere al proceso de asignación de recursos/servicios a tareas, el monitoreo, control y optimización de estados de servicios y recursos y la ejecución de tareas para satisfacer los requisitos individualizados de los consumidores, además del procesamiento de tareas -como la descomposición de ellas, por ejemplo- y el descubrimiento de servicios, entre otros.
6. **Despacho de tareas (Task dispatching):** Un sistema de Cloud Manufacturing debe tener la capacidad de asignar un trabajo a un recurso. Una vez realizada la descomposición de tareas, cada sub tarea debería ser asignada a un servicio, según lo recomendado en la etapa de planificación.
7. **Entrega de resultados (Result delivery):** Una vez finalizada la ejecución del servicio asignado es necesario poder proveer al usuario y al sistema el resultado de dicha ejecución.
8. **Recomendación y descubrimiento de clientes (Customer recommendation and discovery)**
9. **Recomendación y descubrimiento de proveedores/servicios (Provider recommendation and discovery/Service recommendation)**
10. **Generación y mantenimiento de una base de conocimiento:** El planificador hace un fuerte uso del conocimiento adquirido, mientras mayor sea este, mayor serán los resultados obtenidos por dicho planificador, es por ello que es importante contar con una base de conocimiento grande.
11. **Herramientas para manejar y operar el sistema:** Se debe proveer distintas interfaces para que tanto clientes, fabricante como operadores puedan interactuar con el sistema.

## 2.4 Puntos claves de un sistema de CMfg

### Arquitectura

Definir una arquitectura es, sin duda, una tarea muy importante en el desarrollo de un sistema, pero de nada sirve si no es posible implementarla. Dada la complejidad que implica un sistema de Cloud Manufacturing, implementar cada punto de la arquitectura planteada es un desafío y un caso de estudio por sí mismo. Uno de los puntos más estudiados concierne a la planificación de servicios o fabricantes, Alinani, et al. [18] presentan una arquitectura para CMfg descentralizado. Esta arquitectura consiste en cinco puntos claves: Requirement Submission, Task decomposition, Service Scheduling, Generated Service Recommendations y Service Selection/Order.

**Requirement Submission:** El usuario del sistema envía requerimientos con base en varios criterios requeridos para su trabajo, incluyendo el alcance, las capacidades requeridas así como también las tareas y personalizaciones necesarias.

**Task decomposition:** Las tareas requeridas por el usuario son evaluadas y descompuestas en varias sub tareas basadas en su tipo, de modo de asegurar concurrencia y mayor eficiencia. Este paso también es responsable de evaluar la personalización basada en requerimientos funcionales y no funcionales.

**Service Recommender System:** Evalúa las sub tareas descompuestas recibidas por el paso previo de forma de evaluar cada aspecto individualmente para asegurar mayor performance en términos de calidad y tiempo de entrega.  $Rw = Sc + Sq + Lf$ . Donde  $Rw$  es el peso recomendado calculado como la suma de la capacidad del servicio ( $Sc$ ), la calidad de servicio ( $Sq$ ) y la viabilidad logística ( $Lf$ )

**Service Capability:** Usado como criterio de evaluación para cada servicio proporcionados por los proveedores. Este verifica que el proveedor haya realizado tareas con una complejidad y carga de trabajo similar junto con procedimientos y materiales. Con base en estas evaluaciones los servicios se filtran.

**Service Quality:** La evaluación de la calidad del servicio se realiza para garantizar la calidad deseada del producto. Para ello se evalúan los siguientes aspectos: comentarios de los usuarios, entrega oportuna, calidad del producto

**Logistic Feasibility:** Evalúa la posibilidad de obtener un transporte óptimo dentro de la fábrica o en cualquier otra parte del mundo con un tiempo mínimo.

**Generated Service Recommendations:** Elige el mejor servicio basado en su compatibilidad, calidad de los productos entregados, user feedback, factibilidad logística y varios otros criterios.

**Service Selection/Order:** Una vez que las colecciones de servicios recomendados se presentan al usuario, la orden es basada en la selección de servicios de la lista de recomendaciones.

### Registro y Virtualización de recursos

Los fabricantes deben tener la posibilidad de ofrecer sus servicios/recursos a través de la plataforma, quienes a su vez, con el objetivo de mantener controlado el estado del sistema y de poder satisfacer las necesidades de los clientes, deberán ser virtualizados. Existen desafíos interesantes en este sentido, particularmente porque los componentes físicos de los sistemas introducen diferencias cualitativas para con su representación orientada a objetos utilizada normalmente. Se han presentado distintos modelos para representar dichos recursos, uno de ellos son los gemelos digitales, o Digital Twins, que consisten en réplicas digitales de activos físicos potenciales y reales, procesos, personas, lugares, sistemas y dispositivos que pueden utilizarse para diversos fines, todos los activos de las empresas, desde maquinarias hasta servicios. En caso de estar interesado en dicho modelo de datos, en [4] podrá encontrar una documentación más exployada.

### **Presentación de requisitos (Requirement submission)**

Debido a las grandes tecnologías emergente así como también a los cambios constantes en las necesidades de los clientes es necesario poder proporcionar un servicio eficiente que permita satisfacer las distintas necesidades de los consumidores. Alinani, et al [18]. desarrollan dicho paradigma así como también presentan un desarrollo de una plataforma de sistema de configuración de productos para cloud manufacturing. Esta plataforma consta de una arquitectura de 4 capas (capa de instancia, capa de modelo, capa de meta modelo, capa de representación) y se basa fuertemente en OWL para la creación de modelos de información existente, donde OWL es un lenguaje de ontología que permite la reutilización de conocimiento así como también permite compartirlo.

### **Descomposición de tareas (Task decomposition)**

Una tarea, generalmente, puede descomponerse en subtareas con un grado de complejidad inferior. Es altamente útil poder descomponer en sub tareas todas aquellas tareas complejas que no puedan ser desarrolladas por un único servicio.

### **Planificación (Scheduling)**

La planificación es un proceso de organización, control y optimización del trabajo que resulta ser una tarea de vital importancia para CMfg. En el contexto de CMfg se refiere al proceso de asignación de recursos/servicios a tareas, el monitoreo, control y optimización de estados de servicios y recursos y la ejecución de tareas para satisfacer los requisitos individualizados de los consumidores, además del procesamiento de tareas -como la descomposición de ellas, por ejemplo- y el descubrimiento de servicios, entre otros.

La planificación es uno de los temas más abordados por los investigadores en el mundo de Cloud Manufacturing [1, 8, 27, 31]. Hua, et al. [8] presentan un algoritmo basado en chaos optimization algorithm.

Para ello identificaron cinco índices clave:

**Load Task (FZ):** relación entre tareas completadas y no finalizadas

**Manufacturing efficiency (XL):** refleja el grado al cual el fabricante ha logrado su objetivo previsto e indica la relación entre el tiempo de finalización real y el tiempo de finalización prescrito

**Manufacturing resource richness (MR):** proporción de los recursos de fabricación poseídos por el fabricante con los recursos de fabricación requeridos por la tarea

**IoT match degree (LoT) :** capacidades de intercambio de información y logísticas

**Task reliability (RK):** tareas completadas

En dicho paper se desarrollan distintos modelos matemáticos para el cálculo de los índices mencionados así como también para el de su peso, y luego se presenta un planificador de fabricantes basado en Analytic Hierarchy Process y chaos optimization algorithm.

## 2.5 Ejemplos de uso

Según Simeone et al. [28], una industria que muy probablemente se beneficiaría de esto, es la industria de corte de chapa (sheet metal cutting industry), donde los cambios frecuentes de los volúmenes de demanda y variedad de productos, afectan negativamente la eficiencia de los recursos. Reducir el desperdicio de material es posible aplicando una estrategia de anidación (combinar múltiples órdenes de corte de chapa en la misma chapa) CMfg puede ayudar a realizar esto, facilitando la configuración de redes de fabricación conectando clientes y proveedores a lo largo del globo mediante una plataforma común.

El paper “Resource Efficiency Optimization Engine in Smart Production Networks via Intelligent Cloud Manufacturing Platforms” expande el ejemplo de arriba, analizando esta vez desde el punto de vista de los productores. La estrategia para el mejor aprovechamiento de materiales por parte de los productores (minimizar el desperdicio de materiales) se realiza mediante un Genetic Algorithm. La idea es organizar los pedidos de los consumidores de forma tal de utilizar el mayor porcentaje de la placa de chapa. La página 4 del pdf de este paper explica, a alto nivel, cómo funcionaría el motor de apoyo a toma de decisiones.

En dicho paper, se realizó una simulación utilizando dos productores y dos consumidores, mostrando las diferentes formas de organizar el trabajo (incorporar, encolar, priorizar, rechazar consumidores) teniendo en cuenta, además de la maximización del porcentaje de utilización de las materias primas, las deadlines de los consumidores. Adicionalmente, se mostró que el proceso de optimización es costoso en materia de recursos y tiempo de ejecución.

## 2.6 Temas a resolver

### **Definición, descripción, implementación única de idea de Cloud Manufacturing**

No existe aún una definición precisa de qué es CMfg. Se hace necesario tener una definición internacional de CMfg aceptada tanto por la academia como por la industria [2, 7]. Así mismo, durante el proceso de investigación, se detectó que, si bien la mayor parte de los investigadores coinciden en mayor o menor medida en lo que un sistema de Cloud Manufacturing debería hacer, no existe aún una arquitectura, plataforma, modelo, framework o aplicación que implemente CMfg ni se ha estandarizado su diseño. En este sentido, durante la investigación se detectaron varias diferencias en las soluciones que los investigadores han planteado, las cuales van desde distintas arquitecturas hasta distintos planificadores, pasando incluso por distintas representaciones de datos, como VEO [12,13,14] o DT [4].

### **Seguridad de datos e información (en nubes públicas/comunitarias)**

Las empresas no quieren que información confidencial sea accesible por terceros.  
¿Quién es realmente el “dueño” de la parte virtual de un activo de una empresa? ¿La empresa o el administrador de la nube?  
¿Quién es el “dueño” de la información que se obtiene de CMfg?

### **Sociales**

Teniendo en cuenta lo altamente automatizadas que estarían las industrias que trabajen bajo el paradigma de Industrias 4.0, ¿qué roles cumplirían los seres humanos?

### **Estandarización de hardware y software - Integración de sistemas**

Para bajar los costos que conlleva implementar CMfg para una empresa y para asegurar eficiencia y seguridad, debería estandarizarse un protocolo I/O para comunicación entre máquinas y sistemas.

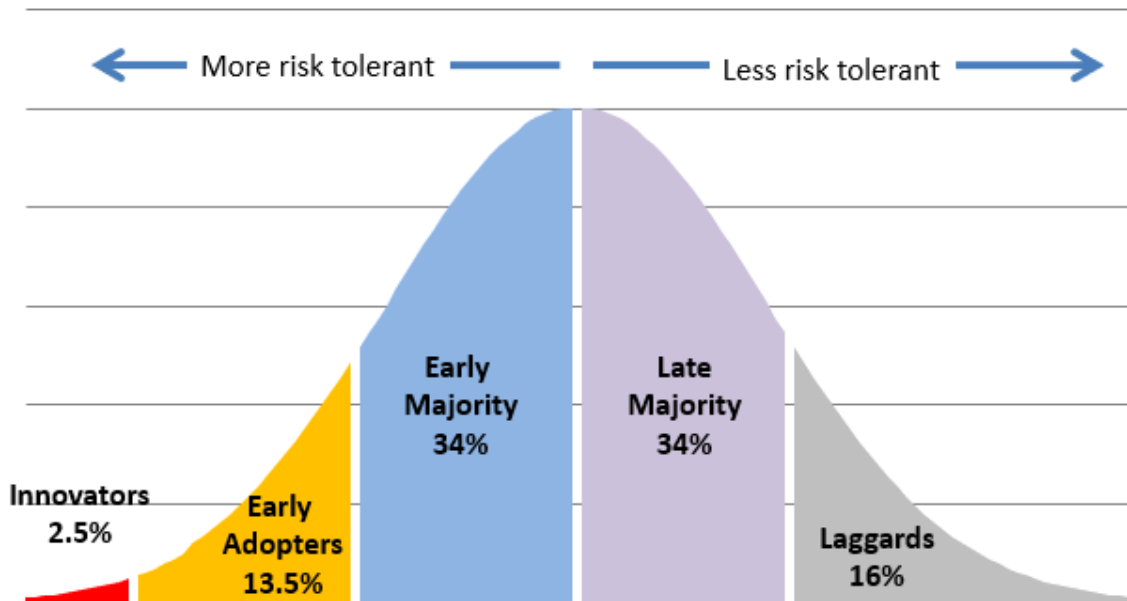
Además, los sistemas de datos y comunicación actuales están diseñados, generalmente, para una correcta comunicación vertical. Hay que reemplazar/modificar estos sistemas para utilizar una comunicación horizontal (por definición misma de CMfg).

### **Adopción - Poblar la nube**

El ciclo de vida de la adopción de tecnologías es un modelo sociológico que puede modelarse como una distribución normal clásica o “curva de campana”<sup>23</sup>, por lo que resulta complejo poblar de usuarios algo nuevo. Por los beneficios que ofrece, una vez que CMfg se encuentre funcionando, se estima que las empresas van a querer ingresar al mundo de las industrias 4.0, pero las primeras empresas -que en la curva de adopción de tecnologías corresponden a los innovadores (innovators) y a los primeros usuarios (early adopters)- deberían ser incentivadas.

---

<sup>23</sup> Fuente: “Basics of The Adoption Curve - Fundamentals of B2B Product Management Series: Part 3”, Michael Mullany (<https://www.linkedin.com/pulse/basics-adoption-curve-michael-mullany>)



**Figura 6:** Porcentaje de clientes que adoptan una innovación a lo largo del tiempo  
Fuente: Mullany (2018) [63]

Se espera que unas de las principales preocupaciones [2] para “pasarse al mundo de CMfg” sean:

1. Seguridad: principalmente los aspectos legales, la privacidad y la protección de datos.
2. Costo: es necesario medir tanto costos como beneficios para demostrar que implementar CMfg resulta rentable para las empresas. La estrategia de “Pago por uso” (“*pay-as-you-go*”) puede resultar muy atractiva.
3. Estrategia de adopción de la nube dentro de la empresa: para obtener el mayor beneficio posible en el uso de CMfg, será necesario contar con una estrategia bien definida y con bases sólidas.

### **Desarrollo de un modelo de negocio para CMfg**

La adopción de CMfg requerirá un nuevo modelo de negocio que apoye la naturaleza colaborativa de CMfg, con socios en red cooperando en la distribución de valor.

## 2.7 Investigación de arquitecturas y plataformas de Cloud Manufacturing

### Primera aproximación: Enterprise Resource Planning en la nube

Los softwares de “Enterprise Resource Planning” (ERP) son sistemas informáticos modulares, configurables y especializados, destinados a la administración de recursos en una organización que automatizan muchas de las prácticas de negocio asociadas a la operativa y producción. Estos sistemas se componen de diferentes módulos que son utilizados para diferentes propósitos, tales como compras, ventas, producción, logística, gestión de proyectos, control de inventarios, entre otros.

En sus inicios, los ERP se enfocaban casi únicamente en satisfacer las necesidades de gestión financiera y administrativa de las organizaciones, sin embargo, en los últimos años, estas herramientas han ido incorporando los procesos de producción dentro de sus competencias.

Algunos de los ERP más completos que se encuentran en la actualidad<sup>24</sup> son Acumatica, Microsoft Dynamics 365 for Finance and Operations, NetSuite ERP, SAP Business One y Epicor ERP.

Si bien este listado de softwares no corresponde a softwares que implementen Cloud Manufacturing tal cual se ha descrito en el informe, son programas relevantes para la investigación que se está llevando a cabo -y también para la industria- en el sentido de que le permiten a las empresas monitorear, automatizar y controlar las diferentes áreas que la componen, a la vez que obtienen datos y generan información que luego es procesada para generar nuevo conocimiento que permita mejorar la gestión.

Estos sistemas, a pesar de la gran distancia que tienen con Cloud Manufacturing, hacen las veces de primera aproximación a la fabricación -y gestión- en la nube.

### Arquitecturas y plataformas relevadas

Para realizar un relevamiento de arquitecturas y plataformas, se realizó una revisión sistemática de literatura para conocer el estado del arte y las investigaciones desarrolladas en temas de arquitecturas y plataformas de Cloud Manufacturing.

Fueron consultadas las bases de datos provistas por los buscadores de IEEE Xplore<sup>25</sup>, Scopus<sup>26</sup> y Springer Link<sup>27</sup>. En los tres casos, se comenzó la búsqueda mediante una consulta inicial en común, “(Cloud Manufacturing AND Industry 4.0) AND (Architectures OR Platforms)”, que fue refinándose con ciertos limitadores y ampliadores según se detalla a continuación.

---

<sup>24</sup> Fuente: “The Best Cloud Manufacturing ERP Software of 2020”, SelectHub (<https://www.selecthub.com/enterprise-resource-planning/myths-cloud-erp-manufacturing>)

<sup>25</sup> IEEE Xplore: <https://ieeexplore.ieee.org/>

<sup>26</sup> Scopus: <https://www.scopus.com/search/form.uri?display=basic>

<sup>27</sup> Springer Link: <https://link.springer.com/>

### **Búsqueda en IEEE Xplore**

- **Query:** (Cloud Manufacturing AND Industry 4.0) AND (Architectures OR Platforms)
- **Content Type:** Journals, Magazines
- **Date published:** 2016 to 2020
- **Total de resultados:** 44

### **Búsqueda de Scopus**

- **Query:** TITLE-ABS-KEY ( ( cloud AND manufacturing AND industry 4.0 ) AND ( architectures OR platforms ) ) AND ( LIMIT-TO ( PUBSTAGE , "final" ) ) AND ( EXCLUDE ( SUBJAREA , "MATH" ) OR EXCLUDE ( SUBJAREA , "PHYS" ) OR EXCLUDE ( SUBJAREA , "BIOC" ) OR EXCLUDE ( SUBJAREA , "CHEM" ) OR EXCLUDE ( SUBJAREA , "ENVI" ) OR EXCLUDE ( SUBJAREA , "EART" ) OR EXCLUDE ( SUBJAREA , "ECON" ) OR EXCLUDE ( SUBJAREA , "MEDI" ) OR EXCLUDE ( SUBJAREA , "NEUR" ) ) AND ( LIMIT-TO ( LANGUAGE , "English" ) ) AND ( LIMIT-TO ( SRCTYPE , "j" ) ) AND ( LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) OR LIMIT-TO ( PUBYEAR , 2018 ) OR LIMIT-TO ( PUBYEAR , 2017 ) OR LIMIT-TO ( PUBYEAR , 2016 ) )
- **Total de resultados:** 71

### **Búsqueda de Springer**

- **Query:** (Cloud Manufacturing AND Industry 4.0) AND (Architectures OR Platforms)
- **Language:** English
- **Subdiscipline:** Manufacturing, Machines, Tools, Processes
- **Content Type:** Article
- **Date published:** 2016 to 2020
- **Exclude:** Preview-Only content
- **Total de resultados:** 59

Las tres búsquedas, realizadas hasta el 31 de diciembre del 2020, retornaron un total de 174 resultados cuya distribución por portal y por año se muestra en Tabla 3 y Gráfica 1 respectivamente.

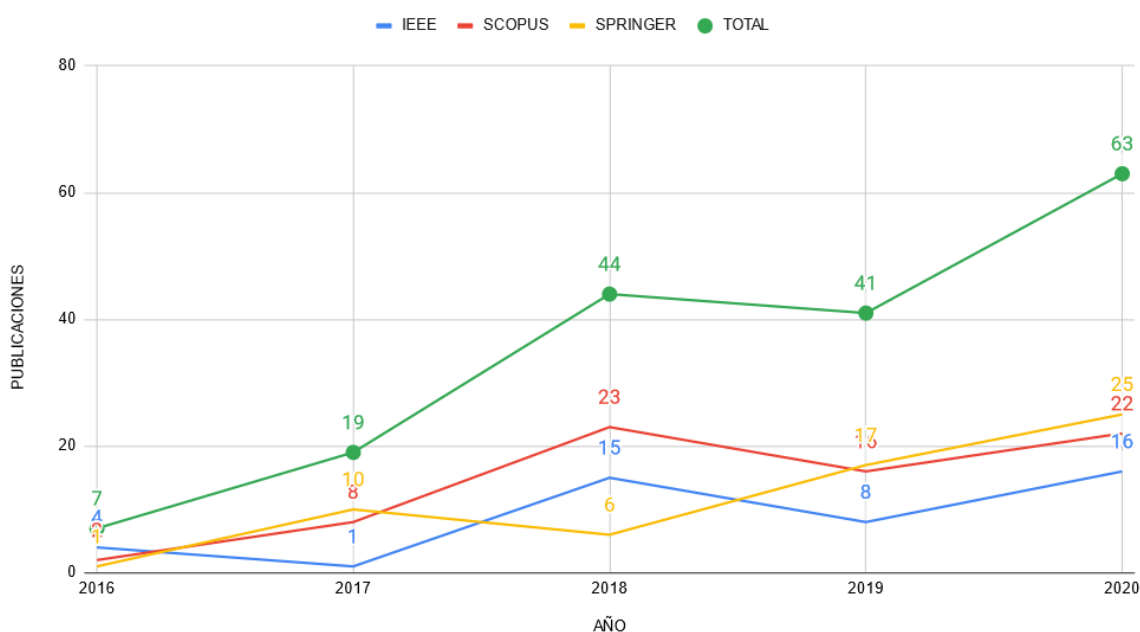
Puede observarse una tendencia creciente en la cantidad de publicaciones por año desde 2016 hasta el año 2020 inclusive.



PORTAL	RESULTADOS	PORCENTAJE
IEEE	44	25.29%
Scopus	71	40.80%
Springer	59	33.91%
<b>TOTAL</b>	<b>174</b>	<b>100.00%</b>

**Tabla 3:** Distribución de resultados de la revisión sistemática por portal  
Fuente: elaboración propia.

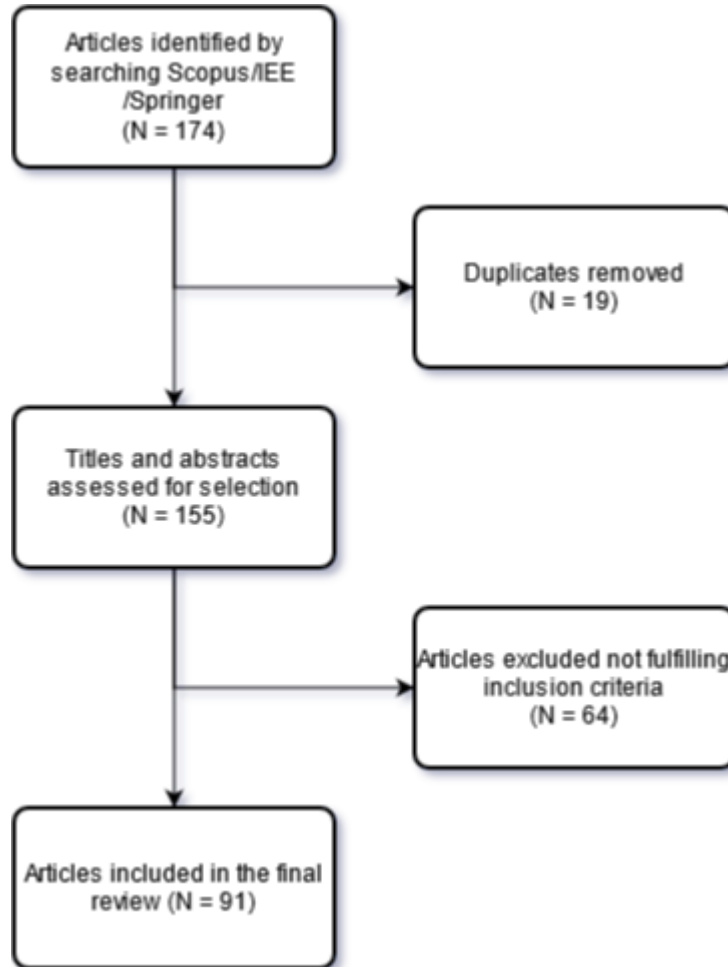
Distribución de publicaciones por año



**Gráfica 1:** Distribución de publicaciones por año  
Fuente: elaboración propia.

Una vez obtenidos los 174 papers, se procedió a detectar y remover resultados duplicados, obteniéndose un total de 19 resultados que fueron retornados en más de un portal.

Una vez alcanzado un resultado de 155 trabajos diferentes, se procedió a realizar un último filtrado manual por título y abstract, que consistió en la lectura de estos fragmentos de papers para definir cuáles eran los trabajos más acordes a los conceptos de arquitecturas y de plataformas en entornos de Cloud Manufacturing y que, por lo tanto, serían más relevantes para esta investigación.



**Diagrama 1:** Diagrama de toma de decisiones  
Fuente: elaboración propia.

El resultado final luego de este último filtro arrojó un resultado de 91 papers, que se encuentran situados entre las posiciones [33, 60] de las referencias (ambos extremos incluidos) que fueron estudiados en detalle y cuyo análisis se refleja a partir de las siguientes líneas.

## 2.8 Resultados y discusión

### **1. Virtualización (Virtualization)**

Uno de los primeros problemas que se encuentran a la hora de implementar un sistema de Cloud Manufacturing consiste en poder conectar el mundo físico con el virtual, para ello es imprescindible contar con una herramienta que permita virtualizar los recursos de una empresa.

Los Cyber Physical Systems (CPS), se definen como tecnologías transformadoras cuyo fin es administrar sistemas interconectados entre sus activos físicos y sus capacidades computacionales [34].

Los sistemas ciber físicos (CPS) han ganado una atención creciente recientemente por su inmenso potencial hacia los sistemas inteligentes de próxima generación que integran la tecnología cibernética en los procesos físicos [49].

La visibilidad y la trazabilidad en tiempo real juegan un papel muy importante en una fábrica inteligente, donde los recursos de fabricación típicos se convierten en objetos de fabricación inteligente (SMO) mediante el uso de varios dispositivos IoT como lectores y etiquetas RFID [35]. Por ello [35] introduce una plataforma de visibilidad y trazabilidad de una fábrica inteligente (iVTP) para así lograr una visualización de la producción en tiempo real de una fábrica inteligente, para ello hace uso de tecnología IoT para identificar varios objetos de fabricación. Específicamente, los dispositivos de identificación por radiofrecuencia (RFID) se utilizan para convertir varios recursos en objetos de fabricación inteligente (SMO) y sus interacciones pueden reflejar en tiempo real las operaciones y comportamientos de producción. A su vez, [34] propone una estructura en 5 niveles de CPS, también llamada 5C-CPS, con el objetivo de proveer una guía paso a paso para desarrollar y desplegar una aplicación de manufacturación en general.

Con respecto a esto fue que se desarrolló el concepto de Digital Twins (DT): una nueva tecnología que ha surgido con el fin de poder virtualizar diferentes elementos del mundo real de una fábrica.

Los Digital Twins han ido tomando mayor relevancia con el pasar de los años, y han captado la atención de muchos investigadores, generando así diversos trabajos que abordan dicho tema. [60], por ejemplo, presenta un estado del arte de DT donde puede encontrarse una revisión de la literatura, las herramientas y modelos principales usados en la creación de los DT, el rol que cumplen el manejo de ciclo de vida del producto y se discuten las ventajas de negocios que estos ofrecen. [61], por su parte, realiza un estudio sobre “Digital Twin based Sustainable Intelligent Manufacturing”, donde destaca que los DT, combinados con un sistema de Intelligent Manufacturing, permiten obtener una producción más inteligente y eficiente. Otros papers que hacen uso de este concepto en diferentes arquitecturas han sido presentados junto a estos trabajos. [62], por ejemplo, propone una arquitectura de 6 capas enfocada en el intercambio de datos e información entre una simulación o emulación remota y su contraparte física mediante el uso de Digital Twin y [37] propone una nueva arquitectura que busca integrar aprendizaje automático así como Digital twin a la estructura 5C-CPS mencionada anteriormente (en [34]) para así lograr un sistema con funcionalidades muy interesantes como self-adaptive, self-learning, y self-maintain.

La tendencia creciente de cantidad de publicaciones sobre este tema en los últimos años indica que serán propuestas más arquitecturas y más aplicaciones que permitan virtualizar recursos de producción. Dos papers que, si bien no hacen énfasis en el concepto de Digital Twin, abordan la misma temática son [36] y [38], donde en [36] se presenta una aplicación de real-time machining data application and service, en esta se adopta la tecnología de fusión multisensor para la adquisición y procesamiento de datos en tiempo real y en [38], se presenta una arquitectura de IIoT definida por software para administrar dispositivos físicos y proporcionan una interfaz para el intercambio de información, tema tratado en la siguiente sección.

## **2. Compartición de recursos (Resources Sharing)**

Además de virtualizar los recursos de producción, un tema importante muy tratado en la literatura es el hecho de compartir recursos (resource sharing).

Compartir los recursos de producción impacta positivamente en la eficiencia de la organización, puesto que se reduce el tiempo que el recurso permanece en desuso, buscando minimizarlo (tratar de reducir al máximo posible la capacidad ociosa).

No se desprende de la investigación que exista actualmente un consenso sobre la forma en la que los recursos de producción puedan ser compartidos eficientemente en ambientes de CMfg, sin embargo, existe una gran cantidad de artículos a este respecto.

Por ejemplo, para habilitar la compartición de un recurso físico ya virtualizado, se propone en [39] utilizar una plataforma inteligente de fabricación en la nube (an intelligent cloud manufacturing platform) para permitir el intercambio dinámico e inteligente de, en este caso, los servicios de corte de chapa (sheet metal cutting services) en una red de fabricación. La solución propuesta, cuyo detalle puede leerse en [39], ofrece soluciones de fabricación inteligente bajo demanda, gestionando y combinando de manera óptima y dinámica, varias instancias de los servicios que son ingresadas (conectadas globalmente a la plataforma común) por clientes y proveedores. Esta plataforma habilitaría la compartición de recursos físicos entre diferentes organizaciones, sin embargo, es posible también compartir recursos -tanto físicos como no físicos- dentro de una misma empresa, como se indica en [42], donde se detalla un sistema de similares características al anterior y se describe una arquitectura estructurada en tres niveles destinada a ofrecer servicios on-demand para evaluar la calidad de productos, ejemplificándose con el caso de uso de la industria aeronáutica.

En este sentido, [45] propone un marco (framework) de fabricación inteligente orientada a servicios (SoSM) para interconexión inteligente de recursos que consta de tres componentes: interconexión dentro de la empresa manufacturera (analizada desde la perspectiva física-física, es decir, entre equipos -o P2P-; ciber-a-ciber, es decir, entre sistemas de información empresariales -o C2C- y física-cibernética, es decir, entre equipos y sistemas de información -o P2C-), interconexión empresarial de la cadena de suministro e interconexión de usuarios en el ciclo de vida del producto.

Por su parte, en [40] se propone, un sistema de fabricación sensible al contexto basado en información, que permite la distribución de la información entre diferentes áreas de una misma industria, donde las áreas de trabajo y los trabajadores ejecutan diferentes tareas y ocupan diferentes puestos durante su turno, este caso conformaría otro ejemplo de compartición de recursos dentro de una misma empresa. En [40], por el contrario, se conforma otro nuevo caso de compartición de recursos entre diferentes empresas. El paper analiza los requisitos -tanto de hardware como de software- de gestión de producción a gran escala orientado a las demandas dinámicas de producción y propone un sistema de servicios de producción en la nube con gestión inteligente dividido en cinco capas bien definidas (physical resource layer, smart resource management layer, iPSS: production resource service layer, PnSS: workshop-level system management & control service layer, S-PnSS enterprise-level synchronization decision & control service layer), cuyo detalle puede verse siguiendo la referencia, que ofrece a los proveedores de recursos una forma de aumentar la frecuencia de arrendamientos de recursos (buscando optimizar la tasa de uso de la plataforma) a la vez que brinda a los clientes soluciones específicas y sistémicas.

Siguiendo la misma línea, la posibilidad de diseño de recursos de fabricación inteligentes que puedan ser conectados e integrados en arquitecturas que permitan convertibilidad, modificabilidad, adaptabilidad y escalabilidad a la vez que, de forma sencilla, puedan integrarse recursos de varios productores y agruparse pequeños servicios para ofrecer un servicio más completo, es estudiada en [41,44], donde el primero realiza una contribución al diseñar una lógica que ataca este problema de capacidad de cambio y reconfiguración de las “fábricas del futuro” haciendo especial foco en el paradigma de total-cost-of-ownership (TCO) y el último presenta una técnica de “service composition and optimal selection” (SCOS) para la creación de un servicio compuesto on-demand de calidad y de forma eficiente. Los cálculos recopilados y realizados pueden verse en dichos trabajos, así como el algoritmo genético orientado a SDF (service domain features) desarrollado por [44]. La composición de servicios es también estudiada por [43], donde se describen de forma general algunas arquitecturas altamente distribuidas basadas en agentes, instanciándolas en una arquitectura en la nube (CASOA), con agentes inteligentes y representaciones ontológicas de la base de conocimiento cuyo objetivo es apoyar la toma de decisiones de los agentes para permitir reconfiguraciones dinámicas, flexibles, ágiles y colaborativas en entornos distribuidos.

La cantidad de capas que conforman las arquitecturas que se plantean también difieren entre los diferentes papers. En [46], por ejemplo, se propone utilizar una arquitectura en tres capas, en lugar de cinco, para compartir los recursos de fabricación entre diferentes clientes en el entorno de fabricación con personalización, teniendo una capa de dispositivos de fabricación (compuesta de varios dispositivos físicos como cintas transportadoras, robots industriales y sensores conectados inalámbricamente a la nube), una capa de sistema de servicio en la nube (donde se procesan los big data de la fabricación) y una capa de servicios móviles (que es utilizada para recibir pedidos de los clientes y para solicitar la información de producción pertinente) y ejemplifica su uso -verificando también la viabilidad del sistema- en un sistema de fabricación de dulces personalizables. El detalle de los experimentos puede verse siguiendo la referencia. [47] realiza un estudio similar pero usando como caso de estudio la industria aeronáutica europea, donde se intenta validar la arquitectura de referencia “SIBUS”, que consiste en un middleware para la adquisición de información, análisis y aplicación en la planificación de recursos.

Finalmente, [48] estudia y evalúa experimentalmente el retraso en la transferencia de información a través de rutas locales e intercontinentales, que resulta también de suma importancia para plataformas de compartición de recursos en la Industria 4.0.

### **3. Planificación (Scheduling)**

Como se ha ido mostrando a lo largo de este documento, en un sistema de Cloud Manufacturing existen múltiples recursos, donde cada uno de ellos pueden llegar a ser utilizados por múltiples dispositivos, a la vez que también existen muchos dispositivos que brindan un mismo servicio, es por ello que es de vital importancia poder administrar todos estos elementos para que puedan trabajar en conjunto y de la forma más optimizada posible sin generar conflictos entre ellos. Es de ahí que surge otro de los problemas más abordados

por la literatura, que consiste en la planificación (scheduling) tanto de recursos, como de servicios.

De la investigación llevada a cabo surgieron diferentes trabajos que atacan esta problemática y a continuación se presenta un resumen de lo encontrado.

Con el fin de resolver la importante cuestión de planificar de forma óptima los recursos agregados, [50] propone una arquitectura de Multi-agent para la planificación en plataformas basadas en smart manufacturing, que consiste en “a platform-level scheduling multi-agent system (MAS) and an enterprise-level scheduling MAS”. Se presentan los procedimientos, características y requerimientos de planificación de una plataforma basada en smart manufacturing y luego se propone un modelo para la planificación basado en la arquitectura propuesta.

Al igual que con las demás problemáticas detectadas para la correcta implementación de la fabricación en la nube donde no existe aún un consenso y los investigadores han propuesto soluciones muy diferentes, la problemática de la planificación no es ajena y también cuenta con un conjunto de propuestas diferentes.

En [54], se propone una arquitectura innovadora “*semi heterarchical*” que permite la integración en diferentes niveles de gestión, cada uno con su cuota de autonomía en la toma de decisiones y responsabilidades específicas (negocios, desempeño general, operativo). Esta arquitectura, aseguran los autores, permite un mejor conocimiento y control del rendimiento del sistema, ofreciendo una mayor previsibilidad de rentabilidad y tiempo de respuesta en el contexto dinámico de Industry 4.0 y Cloud Manufacturing.

En [53] se presenta una arquitectura de fabricación asistida por la nube y de toma de decisiones que contiene una nube y bordes de producción, diseñada con el fin de proveer a los dispositivos tradicionales de la capacidad de acceso a datos y la toma de decisiones propias, [52] describe un framework híbrido y diseña una estrategia de planificación de recursos para cumplir los requerimientos de tiempo real en una fábrica inteligente con edge computing support para lo que en primer lugar se presenta un sistema de cuatro capas en un entorno de fabricación inteligente para soportar la operación de tareas con inteligencia artificial y luego se diseña un algoritmo en dos fases para la planificación de los recursos y [51] presenta un cyber-physical system basado en la nube para una planificación adaptativa en una planta, este trabajo tiene como fin lograr implementar una recopilación, procesamiento y análisis de datos en tiempo real rentables y confiables desde la planta, a la vez que busca usar dichos datos recopilados en un sistema de toma de decisiones adaptativo, que incluye un algoritmo de toma de decisiones de criterios múltiples.

#### **4. Almacenamiento de datos (Data Storage)**

El creciente interés entre los fabricantes por monitorear y analizar sistemas industriales está generando un problema relacionado a los grandes costos asociados con el almacenamiento de los datos obtenidos [55].

Para enfrentar dicho problema, [55] presenta una arquitectura jerárquica de tres niveles para el almacenamiento de series temporales de datos en ambientes en la nube que ayuda a administrar y reducir dichos costos. En el primer nivel, las nuevas series de tiempo sin procesar se pueden almacenar por un corto período en dispositivos electrónicos no volátiles como discos SSD. En el segundo nivel, las series de tiempo recientes se pueden almacenar

durante un período medio en discos duros magnéticos (HDD). En el tercer nivel, una representación reducida de las series de tiempo obtenidas mediante la aplicación de técnicas de reducción de series de tiempo también puede ser almacenada en los discos duros por períodos más prolongados.

La necesidad de mantener datos comunes, a menudo almacenados en ubicaciones remotas tercerizadas y generalmente accesibles para diferentes partes autorizadas, introduce el problema de que estos pueden ser filtrados a entidades no autorizadas. Al utilizar los servicios en la nube, el consumidor puede cifrar datos sensibles antes de cargarlos en la nube, pero esto también eliminará la posibilidad de buscar los datos de manera eficiente en la nube [56]. Para ello [56] propone una variedad de técnicas con el fin de acelerar el rendimiento de búsqueda de datos cifrados almacenados en la nube, estas técnicas incluyen, massively parallel file encryption, multi-array keyword redblack tree (KRBT) implementation, batched keyword search and enhanced parallel search in KRBT.

## **5. Supervisión (Monitoring)**

Resulta también de especial interés el hecho de poder llevar el control de la producción y de la maquinaria en todo momento. El monitoreo de los sistemas de producción y del ciclo de vida de los productos permite, sumado a la recopilación y análisis de grandes volúmenes de información (big data), tomar decisiones sobre la marcha (*on the fly*), ajustar procedimientos, predecir mantenimientos y evitar fallas, lo que redundará en reducción de riesgos y una disminución de costos.

[57] comenta que, si bien existen algunos modelos teóricos cuya precisión para medir actividades es más que aceptable, los que se aplican en la industria suelen quedar rezagados en lo que a recopilación de datos se refiere. La investigación realizada en [57] indica que los métodos de recolección de datos suelen ser manuales y poco precisos. En el susodicho paper se propone un modelo de recopilación de datos en vivo de producción, rechazo y tiempo de inactividad (production, rejection and idle time) apoyándose en sensores electrónicos y IoT. Tanto el detalle como la implementación y validación del modelo (en la fabricación de componentes de enchufes -o plug shell components-) pueden verse en la referencia.

De la misma forma, [58] propone una plataforma a gran escala para el monitoreo en tiempo real de sistemas ciber-físicos (CPS) basado en tecnologías de big data, que es validada en un caso industrial que incluye varias máquinas de prensado industrial.

Como podrá haber observado el lector, el proceso de monitoreo y control resulta una actividad crítica en entornos de Cloud Manufacturing. Además de los trabajos mencionados en las líneas anteriores, existen muchos otros que han realizado investigaciones y han propuesto posibles soluciones a los problemas más comunes. Otro caso interesante que vale la pena mencionar, corresponde al trabajo desarrollado en [59], donde los autores desarrollaron en la nube un entorno para el monitoreo inteligente de procesos de corte con herramientas de IoT. A grandes rasgos, el trabajo consistió en recibir y procesar en tiempo real en la nube datos medidos por sensores que, al final del proceso, son resumidos en un informe automático que incluye el costo y el tiempo total del proceso, así como el consumo promedio de energía en función del tiempo y otras informaciones que puedan ser de

relevancia. El sistema desarrollado en este paper ofrece la posibilidad de que los usuarios puedan visualizar, descargar y analizar la información recopilada, así como proporciona también un algoritmo de predicción para que los usuarios puedan determinar qué tan óptimo es el proceso de producción con respecto al costo y consumo de energía. Trabajos similares se presentan en [32], donde el análisis de datos no se realiza necesariamente en tiempo real; en [29], donde se agrega también un módulo de diagnóstico que activa una acción correctiva local adecuada -como el reemplazo de una herramienta, la detención de un proceso o el cambio de parámetros- ante la detección de algún inconveniente y en [30], que propone una suerte de productos inteligentes, cuyo monitoreo puede realizarse durante todo el ciclo de vida (incluso en poder de los clientes finales) pudiendo solucionar sus problemas mayoritariamente en línea.

Finalmente, papers como [20, 21] también tratan el tema del monitoreo dentro de un sistema de CMfg, desde una visión más general, donde en [21] se hace especial énfasis en que se espera que el monitoreo de los sistemas, sumado a la obtención de datos de sensores vía IoT y el análisis inteligente de los mismos, puedan ser utilizados para tomar decisiones óptimas en todo momento, por ejemplo, a la hora de realizar un control continuo del estado de salud estructural del sistema de fabricación y para facilitar el transporte inteligente, entre otras actividades optimizables.

Adicionalmente, [21] también hace referencia a que la recopilación de datos en estos últimos años donde el costo relativo de almacenamiento suele ser bajo -o al menos no tan elevado- se ha convertido en tendencia y que suelen recopilarse datos incluso antes de que se conozcan las necesidades precisas de los mismos, por lo que insiste en que el monitoreo de sistemas de CMfg, donde la cantidad de sensores que se despliegan va en aumento, debe ser escalable y realizarse de forma ágil y fluida.

## **6. Análisis de datos (Data Analysis)**

Lo abordado en estas últimas páginas lleva, inexorablemente, a la conclusión de que existe una gran importancia en la realización de un correcto análisis de datos para que el desarrollo del Cloud Manufacturing pueda realizarse exitosamente en la industria. Los grandes volúmenes de datos que pueden generarse y recopilarse en cada momento no servirían de nada si no son aprovechados como “materia prima” para una serie de análisis. Es por ello, que no resulta extraño ver una gran cantidad de trabajos de investigación sobre CMfg cuyo eje temático es, justamente, el análisis de datos.

Algunas de las áreas donde resulta aplicable el análisis de la big data en Cloud Manufacturing son el mantenimiento preventivo activo [22, 25, 26], control y optimización de costos de producción [59] y de logística [19], detección y corrección de fallas [29] [30, 25], optimización de procesos [21], predicción de tiempo de vida útil remanente de dispositivos (remaining useful life, RUL) [26], reconfiguración y adaptación mediante re-parametrización óptima [23] entre otros. A medida que la idea de Cloud Manufacturing madura, más aplicaciones podrán aprovechar los resultados de la recopilación y el análisis de datos.

En [22] se propone una arquitectura asistida por la nube para no solo recopilar big data sino utilizar esta información para realizar un mantenimiento activo en tiempo real, así como propone una forma de predicción y análisis “fuera de línea”, a través de dos algoritmos



diferentes -un real-time processing algorithm y un offline prediction algorithm-, cuyo detalle puede verse en la referencia.

[23], por otra parte, estudia la implementación de un módulo de razonamiento con técnicas de aprendizaje automático para atacar el problema de la reconfiguración y adaptación mediante la re-parametrización óptima en sistemas ciberfísicos industriales (reconfiguration and adaptation by means of optimal re-parameterization in Industrial Cyber- Physical Systems, ICPSs) que, según indica, es uno de los cuellos de botella para la transformación digital de la industria manufacturera y propone que el procedimiento de aprendizaje automático se base en dos técnicas de aprendizaje por refuerzo cuya ejecución se realiza en paralelo.

Como se habrá visto en estas líneas, en diferentes artículos se han propuesto diferentes formas de analizar datos para resolver determinado problema, sobran ejemplos de diferentes técnicas utilizadas. Los autores de [24], por ejemplo, presentan en su artículo una plataforma de análisis de datos de procesos sobre plataformas IIoT de última generación, algoritmos de machine learning (ML) para analizar los datos y herramientas de software de big data, mientras que en [26], los autores proponen una suerte de paralelismo entre el monitoreo de las condiciones de la maquinaria industrial con un examen físico para monitorear el estado de salud de los seres humanos: ellos llaman a este concepto “electrocardiograma de dispositivo” (device electrocardiogram DEDG), apoyado con técnicas de deep learning.

Por otra parte, en [19], lo que se busca es optimizar el transporte de productos para minimizar los tiempos de entrega y reducir lo máximo posible los costos de transporte. El trabajo realizado en dicho artículo -que además propone un modelo de fabricación de máscaras faciales personalizadas y sostenibles- indica que el proceso de optimización de transporte, que consta del procesamiento en la nube del big data del tráfico a través de un módulo de logística inteligente, puede ser aplicable a pequeñas y medianas empresas (PYMEs -o Small and Medium Enterprises, SME- en inglés).

Finalmente, vale la pena nombrar la investigación realizada en [25] donde utilizan tecnologías como data lake, bases de datos NoSQL, Apache Spark<sup>28</sup> (motor de análisis para procesamiento de datos a gran escala), Apache Drill<sup>29</sup> (motor de consultas SQL sin esquema para Hadoop, NoSQL y Cloud Storage), Apache Hive<sup>30</sup> (software de almacenamiento de datos), OPC Collector para construir una arquitectura que supere los desafíos de ingestión, integración, transformación, almacenamiento, análisis y visualización de big data en tiempo real.

Muchos de estos trabajos se apoyan en los conceptos de Edge y Fog Computing, que consisten en una estructura de red descentralizada en la que los recursos se sitúan en algún lugar lógico entre la nube y la fuente que genera los datos, para diseñar las diferentes formas de recopilar y analizar los datos en contextos de Cloud Manufacturing. La ventaja más notoria que tiene aplicar Edge y Fog Computing en CMfg se da en la mejora de

---

<sup>28</sup> Apache Spark: <https://spark.apache.org/>

<sup>29</sup> Apache Drill: <https://drill.apache.org/>

<sup>30</sup> Apache Hive: <https://hive.apache.org/>

eficiencia que conlleva el hecho de que los servicios que analizan los datos se encuentren más cerca del origen de los mismos, reduciendo la distancia que deben recorrer los datos, la latencia y haciendo un menor uso de la red.

## **7. Seguridad y optimizaciones (Security & Optimizations)**

Aunque los sistemas de la industria 4.0 tienen la oportunidad de mejorar la productividad y la rentabilidad de las organizaciones, aún enfrentan grandes desafíos relacionados con la ciberseguridad y la privacidad, conceptos como la transparencia de datos que juegan un factor muy importante en la industria 4.0 resultan en consecuencias potencialmente importantes para los ciberataques exitosos [11].

En [11] se propone un nuevo esquema de seguridad “*threat intelligence scheme*” que modela las interacciones dinámicas de los componentes de la industria 4.0, incluyendo tanto sistemas físicos como sistemas en la red. El esquema consiste en dos componentes, “*a smart management module and a threat intelligence module*”. El primero es el encargado de manejar las fuentes de datos, mientras que el segundo se encarga de descubrir actividades anómalas contra los sistemas físicos y de la red. Otro artículo que aborda temas de ciberseguridad es [10], que presenta un framework de ciberseguridad y resiliencia para aplicaciones de fabricación basadas en redes definidas por software, el consta de un modelo combinado de ontología de resiliencia - ciberseguridad y de mecanismos para mantener el sistema con el nivel de seguridad requerido.

Avanzando hacia temas de optimizaciones se encuentra el trabajo realizado en [16], que propone una nueva arquitectura para el procesamiento de grandes cantidades de datos llamada Phi architecture que es capaz de realizar varias funciones como batch data processing y stream data processing, acceso y almacenamiento distribuido, y control en tiempo real. Esta nueva arquitectura está diseñada para un patrón de microservicios, mejorando de esta forma la flexibilidad y estabilidad de la arquitectura.

Otra forma que se encontró en [15] para optimizar los sistemas consiste en un enfoque sistemático y comprensivo, basado en una arquitectura de tres niveles, cuyo fin es comprender modelos predictivos de procesos de fabricación a partir de fuentes dispares de información, analíticas, y experimentales. Su principal ventaja es que facilita la toma de decisiones complejas basadas en múltiples criterios de decisión.

El tamaño de la empresa también juega un papel clave a la hora de poder realizar ciertas optimizaciones en el proceso de fabricación. Las pequeñas y medianas empresas a menudo se ven obstaculizadas a la hora de aprovechar al máximo las últimas tecnologías como lo son IoT, la nube y big data, esto se debe a los grandes costos que estos implican. Para una empresa que opera un sistema de información de manufactura, la administración integrada de información entre sistemas es esencial para la aplicación de una nueva tecnología. Si la empresa carece de los expertos pertinentes, tendrá dificultades para aplicar una nueva tecnología en el campo [17]. Por ello [17] sugiere el uso de una plataforma de operación e integración de aplicaciones basada en la nube (applications' integration and operation platform - AIOP) para resolver dichos problemas, esta aplicación debe ser capaz de aceptar grandes volúmenes de datos en los campos de fabricación basados en IoT, interconectarse

entre los campos de fabricación y AIOP, y proporcionar contenidos de aplicaciones en forma de servicios.

La industria 4.0 introduce una amplia gama de nuevos requisitos para las redes industriales que demandan las aplicaciones que interactúan en los servicios en la nube, dando como resultado una amplia gama de desafíos en cuanto a la gestión y control de la red. Las tecnologías actuales no se adecuan bien a escenarios con diversas tecnologías de comunicación [3].

[3] con el fin de resolver estos desafíos considera la división de redes y presenta métodos para dividir los protocolos de comunicación industriales deterministas y conmutados por paquetes, que simplifican la capacidad de administración de redes heterogéneas con diversos requisitos de aplicación.

El manejo de materiales es una parte esencial de una fábrica inteligente, por ello [6] presenta y analiza una entidad industrial cognitiva llamada context aware cloud robotics (CACR). Esta se caracteriza por dos características, primero, por servicios sensibles al contexto y segundo, por el balanceo de carga efectivo.

## Plataformas para Industrias 4.0

De la investigación realizada anteriormente -en la revisión sistemática de artículos sobre arquitecturas de fabricación en la nube- surgió también la existencia de una plataforma que permite, a través de SDKs para diferentes lenguajes de programación, la implementación de muchos de los conceptos que serán utilizados en sistemas para industrias 4.0.

En este sentido, se encontró de especial interés realizar una investigación acerca de esta nueva plataforma, que se describe en las líneas que siguen a continuación.

### BaSys 4

BaSys 4 consiste en un middleware que define una arquitectura de referencia para los sistemas de producción que permite el cambio a la Industria 4.0.

El proyecto BaSys 4.0 comenzó el 1° de julio del año 2016 y se extendió durante tres años hasta el 2019. Este proyecto fue financiado por el Ministerio Federal de Educación e Investigación (BMBF) de Alemania con 12 millones de euros.

La primera etapa del proyecto BaSys fue calificada como exitosa y por dicho motivo, el BMBF financió con 11,3 millones de euros su segunda parte, BaSys 4.2, durante los siguientes tres años. Actualmente, corriendo el año 2020, se continúa trabajando en este último proyecto.

### Eclipse BaSyx

Eclipse BaSyx es una implementación de referencia de los conceptos de BaSys 4.0 y BaSys 4.2 en una plataforma de código abierto desarrollada por BaSys en conjunto con la Fundación Eclipse.

Los conceptos desarrollados en el proyecto BaSyx se implementan y ponen a disposición en forma de kits de desarrollo de software de código abierto (SDK) para diferentes lenguajes de programación.

A continuación, se listan de forma general las características principales del proyecto y su relevancia para el desarrollo de sistemas para la Industria 4.0.

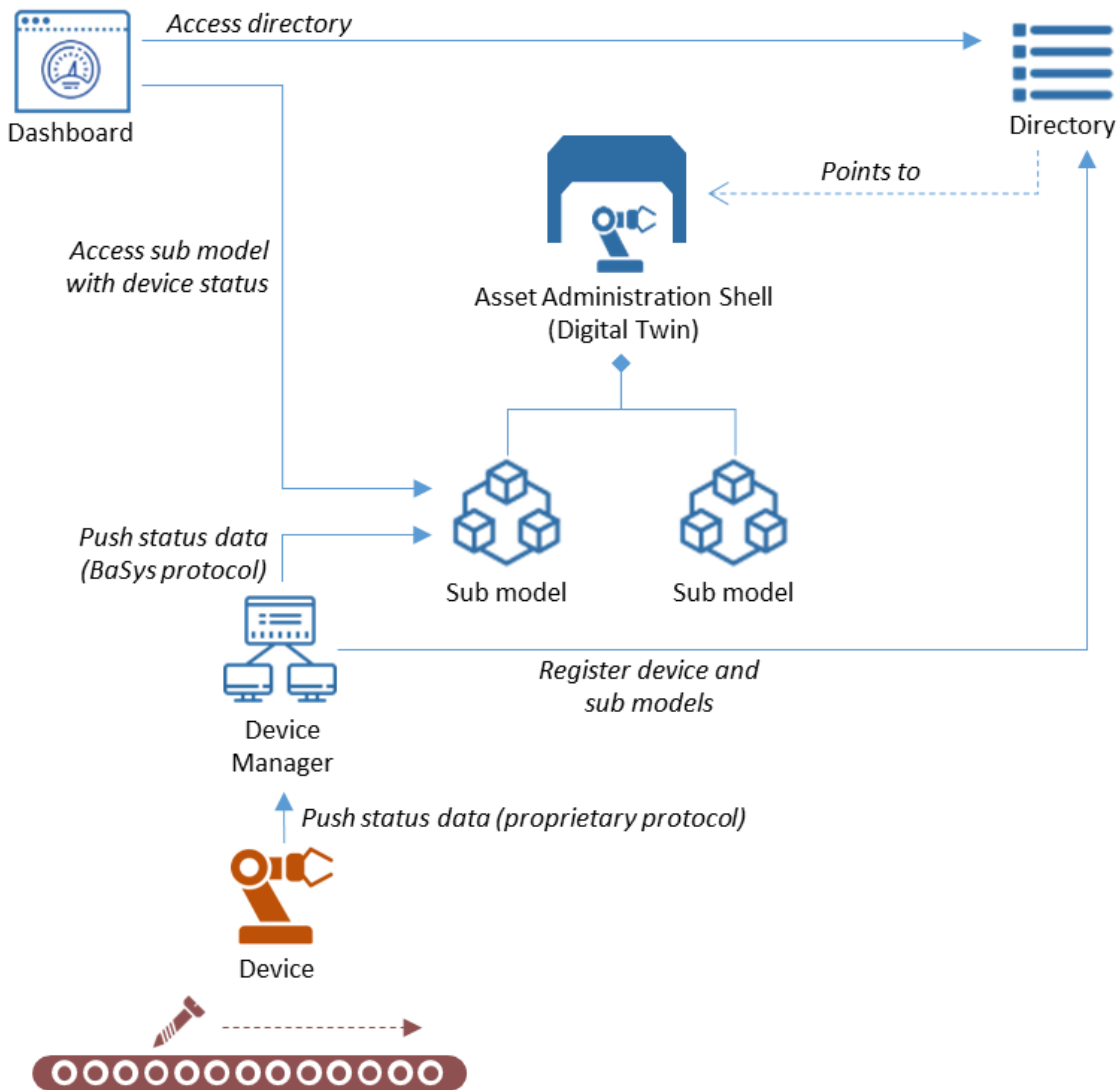
#### Modelado de Objetos

BaSyx provee la capacidad de poder representar elementos claves de la industria 4.0. Para ello, provee dos clases fundamentales: Asset y Asset Administration Shell (AAS).

La clase Asset permite modelar objetos de la realidad (activos). Gracias a ella es posible crear Digital Twins para encapsular toda la información relevante asociada al activo que se quiere representar, así como también los distintos servicios que ofrece dicho activo.

Los Assets son encapsulados en Asset Administration Shells (AAS), como muestra la Figura 7, que proveen la interfaz para hacer uso de los distintos servicios asociados a dicho asset. Para ello, la clase AAS ofrece la posibilidad de crear distintos submódulos donde publicar distintos servicios provistos por el Asset, permitiendo, de esta forma, realizar una interacción con el objeto representado a través de un conjunto organizado de servicios y propiedades.



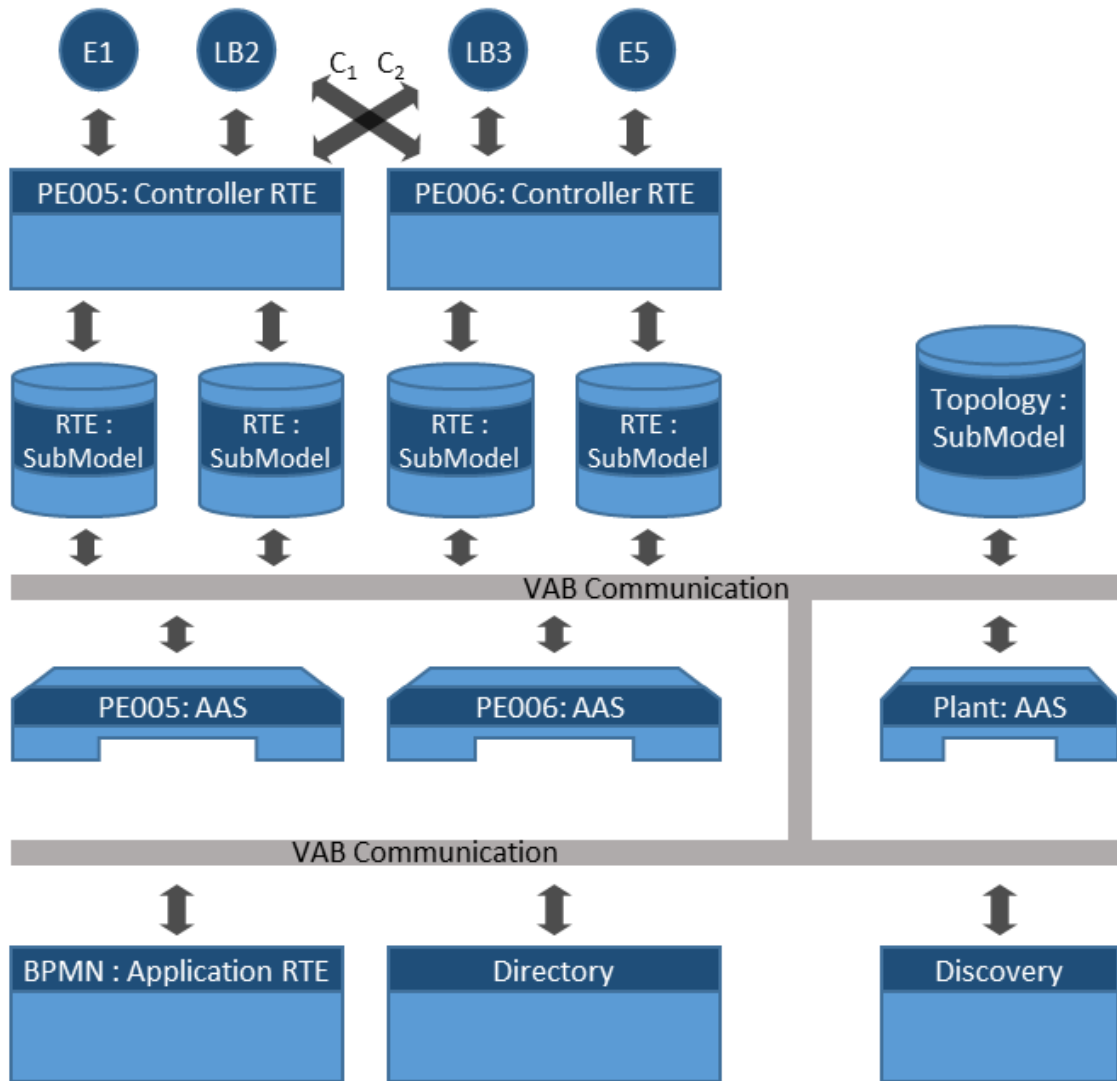


**Figura 8:** ejemplo de arquitectura de Asset Administration Shell en BaSyx  
Fuente: Kuhn (2019) [65]

Hoy en día existen una gran variedad de aparatos tecnológicos que implementan herramientas de IoT, donde cada uno de ellos utiliza protocolos para conectarse a la red. Dado que no todos estos dispositivos implementan el mismo protocolo, surge la necesidad de poder comunicarse con distintos protocolos. Para ello BaSyx provee una herramienta llamada Virtual Automation Bus (VAB), cuyo fin es proveer una comunicación end to end para la industria 4.0.

Esta implementación permite mapear una semántica de comunicación con cinco primitivas (create, delete, retrieve, update, invoke) a diferentes redes y protocolos.

Cada protocolo puede asignarse al VAB mediante la creación de una asignación de estas cinco primitivas a sus respectivas técnicas de comunicación nativas. Con esto, se reduce el esfuerzo necesario para adaptar distintos tipos de tecnologías de comunicación.



**Figura 9:** concepto de Virtual Automation Bus (VAB) con entidades conectadas  
Fuente: Kuhn (2018) [66]

La Figura 9 muestra un diagrama de concepto de VAB con algunas entidades conectadas, donde:

- Los Controller runtime environments (Controller RTE) son entornos de ejecución en tiempo real que son especialmente diseñados para el control automático. Estos implementan, por ejemplo, controladores de dispositivos que proveen acceso basado en servicios a dispositivos controlados. En el ejemplo mostrado en la Figura 9 estos serían motores actuadores (E1 y E5), así como barreras livianas (LB2 y LB3)
- Application runtime environments (Application RTE), ejecuta aplicaciones de alto nivel las cuales interactúan con la planta. A su vez también implementan componentes de grupos, los cuales realizan servicios de alto nivel basados en servicios de otro componente de grupo o dispositivo. Las aplicaciones RTE contienen aplicaciones que ofrecen y utilizan llamadas de servicio para interactuar

con otras funciones. Requieren acceso al VAB para comunicarse con los dispositivos de la planta.

- El discovery service es una forma unificada para que los componentes se conecten con entidades VAB y descubran AAS.
- Los elementos VAB proporcionan una interfaz unificada para acceder a las entidades que están conectadas al VAB. Admiten el acceso de lectura / escritura a los valores e implementan la operación invocable, así como la creación dinámica opcional y la eliminación de propiedades.



### 3. Conclusiones

Las plataformas de fabricación en la nube surgieron como un paradigma de producción que permite explotar todo el potencial de las nuevas tecnologías emergentes y sus avances, llevando así, a lo que podría considerarse como la cuarta revolución industrial.

Mediante la presente investigación, se observó que las áreas de virtualización de recursos y de datos han sido los campos más estudiados en la literatura revisada. Esto insinúa que las necesidades básicas de CMfg serían tener la posibilidad de ofrecer recursos de producción en una plataforma virtual para mostrar su potencial, estado y capacidades en la plataforma.

Además, el análisis de datos es otro aspecto fundamental en las arquitecturas CMfg, ya que, mediante el análisis de los datos de los proveedores de recursos productivos y de los requerimientos de los clientes, es que CMfg cumple con su función de enlace.

En cuanto a otros aspectos de las arquitecturas CMfg, también se han estudiado, aunque con menor intensidad, la compartición de recursos, la planificación y logística, y la monitorización.

De estas tres áreas, cabe señalar que a medida que se consoliden los desarrollos y avances, CMfg irá aumentando la probabilidad de desarrollo y adopción por nuevos nichos comerciales.

Por otro lado, se necesita un estudio más profundo en las áreas de *Digital Twins* y composición de los servicios de producción para permitir un mejor uso de las arquitecturas CMfg.

En cuanto a los *Digital Twins*, cuanto mejor sea la representación virtual de los procesos físicos y las capacidades productivas, mejores decisiones se podrán tomar desde la plataforma de CMfg. Por otro lado, una mejor composición de los servicios permitirá ofrecer un diferencial con respecto a otros paradigmas productivos, ya que el objetivo de CMfg es reunir distintos proveedores de recursos con diferentes capacidades en un solo sistema. Por tanto, desde una visión integradora de ellos, es posible lograr una sinergia entre estas capacidades, ofreciendo servicios integrados efectivamente.

Finalmente, se proporcionó una descripción del proyecto de código abierto BaSys y su implementación BaSyx, que tiene las capacidades para facilitar y mejorar la adopción de arquitecturas de CMfg.

## Glosario

La presente glosa propone definir o comentar una serie de conceptos generales de la temática de fabricación en la nube con el fin de clarificarlos para apoyar al lector en el entendimiento del trabajo.

Las definiciones y comentarios de este glosario que son tomadas del Diccionario de la lengua española de la Real Academia Española se encuentran identificados con la etiqueta [R.A.E.], mientras que las definiciones y comentarios tomadas de Wikipedia se encuentran identificados con la etiqueta [Wikipedia].

**Asset:** cualquier activo físico, proceso, persona, lugar, sistema o dispositivo.

**Algoritmo:** conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. [R.A.E.]

**Big data:** conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales. [Libro “Nuevas Tecnologías y Globalización”, Pavel Sidorenko Bautista, Editorial Elearning, ISBN: 978-84-17814-54-0]

**Digital Twin (DT):** réplicas digitales de activos físicos potenciales y reales, procesos, personas, lugares, sistemas y dispositivos que pueden utilizarse para diversos fines, todos los activos de las empresas, desde maquinarias hasta servicios.

**CMfg:** Cloud Manufacturing

**Cloud:** la nube, o Cloud, es una metáfora que se utiliza para designar una red mundial de servidores remotos, cada uno con una función única, que funciona como un único ecosistema, normalmente asociada a Internet. Estos servidores están diseñados para almacenar y administrar datos, ejecutar aplicaciones o entregar contenido o servicios.

**Cloud Computing (CC):** paradigma que permite ofrecer servicios de computación a través de una red, la cual suele ser internet. [Wikipedia]

**Everything as a Service (XaaS):** consiste en una gran variedad de servicios y productos que pueden ser accedidos por los usuarios bajo demanda a través de internet.

**Holismo; Holístico, ca:** posición metodológica y epistemológica que postula cómo los sistemas (físicos, biológicos, sociales, económicos, mentales, lingüísticos, etc.) y sus propiedades deben ser analizados en su conjunto y no sólo a través de las partes que los componen. [Wikipedia]

**Internet of Things (IoT):** concepto que se ha puesto de moda en los últimos años, consiste en un sistema de dispositivos interrelacionados con la capacidad de comunicarse entre ellos sin intervención humana. Es, en definitiva, la conexión de internet más con objetos que con personas. [Wikipedia]

**Matching:** Vinculación o emparejamiento entre dos o más elementos. En el contexto de este trabajo, por ejemplo, entre actores/usuarios del sistema, o entre un requerimientos y recursos de fabricación.

**Middleware:** software que proporciona un enlace entre aplicaciones de software independientes. Se utiliza para simplificar el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones entre sistemas distribuidos, redundando en una mejora de la calidad de servicio, la seguridad, etcétera. [Wikipedia]

**Nube:** ver definición en **Cloud**.

**Ontología:** una *ontología* consiste en una definición formal de tipos, propiedades, y relaciones entre entidades. [Wikipedia]

**Paradigma:** teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento. [R.A.E.]

**Pool:** se le llama *pool* virtual de servicios a un grupo o conjunto de servicios que se encuentran disponibles dentro de un sistema.

**Pyme:** acrónimo de pequeña y mediana empresa. Empresa mercantil, industrial, etc., compuesta por un número reducido de trabajadores, y con un moderado volumen de facturación. [R.A.E.]

**Sistema ciber-físico:** mecanismo (sistema físico) controlado o monitorizado por algoritmos basados en computación y estrechamente integrados con Internet. [Wikipedia]

**Software:** conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. [R.A.E.]

## Referencias

- [1] Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, 57(15-16), 4854-4879.
- [2] Adamson, G., Wang, L., Holm, M., & Moore, P. (2017). Cloud manufacturing—a critical review of recent development and future trends. *International Journal of Computer Integrated Manufacturing*, 30(4-5), 347-380.
- [3] Kalør, A. E., Guillaume, R., Nielsen, J. J., Mueller, A., & Popovski, P. (2018). Network slicing in industry 4.0 applications: Abstraction methods and end-to-end analysis. *IEEE Transactions on Industrial Informatics*, 14(12), 5419-5427.
- [4] Dietz, M., & Pernul, G. (2020). Digital twin: Empowering enterprises towards a system-of-systems approach. *Business & Information Systems Engineering*, 62(2), 179-184
- [5] Ellwein, C., Neff, S., & Verl, A. (2019). Cloud Manufacturing: An Automated Literature Review. *Procedia CIRP*, 86, 251-256.
- [6] Wan, J., Tang, S., Hua, Q., Li, D., Liu, C., & Lloret, J. (2017). Context-aware cloud robotics for material handling in cognitive industrial Internet of Things. *IEEE Internet of Things Journal*, 5(4), 2272-2281.
- [7] Henzel, R., & Herzwurm, G. (2018). Cloud Manufacturing: A state-of-the-art survey of current issues. *Procedia CIRP*, 72, 947-952.
- [8] Hu, Y., Zhu, F., Zhang, L., Lui, Y., & Wang, Z. (2019). Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 58, 13-20.
- [9] Ren, L., Zhang, L., Tao, F., Zhao, C., Chai, X., & Zhao, X. (2015). Cloud manufacturing: from concept to practice. *Enterprise Information Systems*, 9(2), 186-209.
- [10] Babiceanu, R. F., & Seker, R. (2019). Cyber resilience protection for industrial internet of things: A software-defined networking approach. *Computers in Industry*, 104, 47-58.
- [11] Moustafa, N., Adi, E., Turnbull, B., & Hu, J. (2018). A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access*, 6, 32910-32924.
- [12] Shafiq, S. I., Sanin, C., Szczerbicki, E., & Toro, C. (2014). Implementing virtual engineering objects (VEO) with the set of experience knowledge structure (SOEKS). *Procedia Computer Science*, 35, 644-652.

- [13] Shafiq, S. I., Sanin, C., Szczerbicki, E., & Toro, C. (2015). Virtual engineering object/virtual engineering process: a specialized form of cyber physical system for Industrie 4.0. *Procedia Computer Science*, 60, 1146-1155.
- [14] Shafiq, S. I., Sanin, C., Toro, C., & Szczerbicki, E. (2016). Virtual engineering process (VEP): a knowledge representation approach for building bio-inspired distributed manufacturing DNA. *International Journal of Production Research*, 54(23), 7129-7142.
- [15] Kim, D. B. (2019). An approach for composing predictive models from disparate knowledge sources in smart manufacturing environments. *Journal of Intelligent Manufacturing*, 30(4), 1999-2012.
- [16] Wang, W., Fan, L., Huang, P., & Li, H. (2019). A New Data Processing Architecture for Multi-Scenario Applications in Aviation Manufacturing. *Ieee Access*, 7, 83637-83650.
- [17] Jun, C., Lee, J. Y., Yoon, J. S., & Kim, B. H. (2017). Applications' integration and operation platform to support smart manufacturing by small and medium-sized enterprises. *Procedia Manufacturing*, 11, 1950-1957.
- [18] Alinani, K., Liu, D., Zhou, D., & Wang, G. (2019, November). Recommender system for decentralized cloud manufacturing. In *International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications* (pp. 170-179). Springer, Singapore.
- [19] Liu, C., Zhou, Y., Cen, Y., & Lin, D. (2019). Integrated application in intelligent production and logistics management: technical architectures concepts and business model analyses for the customised facial masks manufacturing. *International Journal of Computer Integrated Manufacturing*, 32(4-5), 522-532.
- [20] Liu, Y., Tong, K., Mao, F., & Yang, J. (2019). Research on digital production technology for traditional manufacturing enterprises based on industrial Internet of Things in 5G era. *The International Journal of Advanced Manufacturing Technology*, 1-14.
- [21] Ooi, B. Y., Kong, Z. W., Lee, W. K., Liew, S. Y., & Shirmohammadi, S. (2019). A collaborative IoT-gateway architecture for reliable and cost effective measurements. *IEEE Instrumentation & Measurement Magazine*, 22(6), 11-17.
- [22] Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4), 2039-2047.
- [23] Villalonga, A., Beruvides, G., Castano, F., & Haber, R. E. (2020). Cloud-based industrial cyber-physical system for data-driven reasoning: A review and use case on an industry 4.0 pilot line. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 16(9), 5975-5984.
- [24] Kabugo, J. C., Jämsä-Jounela, S. L., Schiemann, R., & Binder, C. (2020). Industry 4.0 based process data analytics platform: A waste-to-energy plant case study. *International Journal of Electrical Power & Energy Systems*, 115, 105508.

- [25] Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2019). A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics*, 16(1), 183-192.
- [26] Yan, H., Wan, J., Zhang, C., Tang, S., Hua, Q., & Wang, Z. (2018). Industrial big data analytics for prediction of remaining useful life based on deep learning. *IEEE Access*, 6, 17190-17197.
- [27] Mourtzis, D., Doukas, M., Lalas, C., & Papakostas, N. (2015). Cloud-based integrated shop-floor planning and control of manufacturing operations for mass customisation. *Procedia CIRP*, 33, 9-16.
- [28] Simeone, A., Caggiano, A., Deng, B., & Boun, L. (2019). A deep learning based-decision support tool for solution recommendation in cloud manufacturing platforms. *Procedia CIRP*, 86, 68-73.
- [29] Caggiano, A. (2018). Cloud-based manufacturing process monitoring for smart diagnosis services. *International Journal of Computer Integrated Manufacturing*, 31(7), 612-623.
- [30] Kuru, K., & Yetgin, H. (2019). Transformation to advanced mechatronics systems within new industrial revolution: A novel framework in automation of everything (AoE). *IEEE Access*, 7, 41395-41415.
- [31] Rossit, D., & Tohmé, F. (2018). Scheduling research contributions to Smart manufacturing. *Manufacturing Letters*, 15, 111-114.
- [32] Mourtzis, D., Milas, N., & Vlachou, A. (2018). An internet of things-based monitoring system for shop-floor control. *Journal of Computing and Information Science in Engineering*, 18(2).
- [33] Zhou, Z., Hu, J., Liu, Q., Lou, P., Yan, J., & Li, W. (2018). Fog computing-based cyber-physical machine tool system. *IEEE Access*, 6, 44580-44590.
- [34] Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, 3, 18-23.
- [35] Zhong, R. Y., Xu, X., & Wang, L. (2017). IoT-enabled smart factory visibility and traceability using laser-scanners. *Procedia Manufacturing*, 10, 1-14.
- [36] Tong, X., Liu, Q., Pi, S., & Xiao, Y. (2019). Real-time machining data application and service based on IMT digital twin. *Journal of Intelligent Manufacturing*, 1-20.
- [37] Lee, J., Azamfar, M., Singh, J., & Siahpour, S. (2020). Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing. *IET Collaborative Intelligent Manufacturing*, 2(1), 34-36.

- [38] Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. V. (2016). Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*, 16(20), 7373-7380.
- [39] Simeone, A., Deng, B., & Caggiano, A. (2020). Resource efficiency enhancement in sheet metal cutting industrial networks through cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, 1-21.
- [40] Zhang, K., Wan, M., Qu, T., Jiang, H., Li, P., Chen, Z., ... & Huang, G. Q. (2019). Production service system enabled by cloud-based smart resource hierarchy for a highly dynamic synchronized production process. *Advanced Engineering Informatics*, 42, 100995.
- [41] Brad, S., Murar, M., & Brad, E. (2018). Design of smart connected manufacturing resources to enable changeability, reconfigurability and total-cost-of-ownership models in the factory-of-the-future. *International Journal of Production Research*, 56(6), 2269-2291.
- [42] Caggiano, A., Segreto, T., & Teti, R. (2020). Cloud manufacturing architecture for part quality assessment. *Cogent Engineering*, 7(1), 1715524.
- [43] Tang, H., Li, D., Wang, S., & Dong, Z. (2017). CASOA: an architecture for agent-based manufacturing system in the context of industry 4.0. *IEEE Access*, 6, 12746-12754.
- [44] Li, T., He, T., Wang, Z., & Zhang, Y. (2020). SDF-GA: a service domain feature-oriented approach for manufacturing cloud service composition. *Journal of Intelligent Manufacturing*, 31(3), 681-702.
- [45] Tao, F., & Qi, Q. (2017). New IT driven service-oriented smart manufacturing: framework and characteristics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 81-91.
- [46] Wan, J., Yi, M., Li, D. I., Zhang, C., Wang, S., & Zhou, K. (2016). Mobile services for customization manufacturing systems: An example of industry 4.0. *IEEE Access*, 4, 8977-8986.
- [47] Yoon, S., Um, J., Suh, S. H., Stroud, I., & Yoon, J. S. (2019). Smart Factory Information Service Bus (SIBUS) for manufacturing application: requirement, architecture and implementation. *Journal of Intelligent Manufacturing*, 30(1), 363-382.
- [48] Ferrari, P., Flammini, A., Sisinni, E., Rinaldi, S., Brandão, D., & Rocha, M. S. (2018). Delay estimation of industrial IoT applications based on messaging protocols. *IEEE Transactions on Instrumentation and Measurement*, 67(9), 2188-2199.
- [49] Yao, X., Zhou, J., Lin, Y., Li, Y., Yu, H., & Liu, Y. (2019). Smart manufacturing based on cyber-physical systems and beyond. *Journal of Intelligent Manufacturing*, 30(8), 2805-2817.

- [50] Liu, Y. K., Zhang, X. S., Zhang, L., Tao, F., & Wang, L. H. (2019). A multi-agent architecture for scheduling in platform-based smart manufacturing systems. *Frontiers of Information Technology & Electronic Engineering*, 20(11), 1465-1492.
- [51] Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of manufacturing systems*, 47, 179-198.
- [52] Li, X., Wan, J., Dai, H. N., Imran, M., Xia, M., & Celesti, A. (2019). A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing. *IEEE Transactions on Industrial Informatics*, 15(7), 4225-4234.
- [53] Tang, H., Li, D., Wan, J., Imran, M., & Shoaib, M. (2019). A reconfigurable method for intelligent manufacturing based on industrial cloud and edge intelligence. *IEEE Internet of Things Journal*, 7(5), 4248-4259.
- [54] Grassi, A., Guizzi, G., Santillo, L. C., & Vespoli, S. (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*.
- [55] Villalobos, K., Ramírez-Durán, V. J., Diez, B., Blanco, J. M., Goñi, A., & Illarramendi, A. (2020). A three level hierarchical architecture for an efficient storage of industry 4.0 data. *Computers in Industry*, 121, 103257.
- [56] Lee, W. K., Phan, R. C. W., Poh, G. S., & Goi, B. M. (2018). SearchaStore: fast and secure searchable cloud services. *Cluster Computing*, 21(2), 1189-1202.
- [57] BA, P., PN, S., & PM, S. (2020). Shop floor to cloud connect for live monitoring the production data of CNC machines. *International Journal of Computer Integrated Manufacturing*, 33(2), 142-158.
- [58] Canizo, M., Conde, A., Charramendieta, S., Minon, R., Cid-Fuentes, R. G., & Onieva, E. (2019). Implementation of a large-scale platform for cyber-physical system real-time monitoring. *IEEE Access*, 7, 52455-52466.
- [59] Giraldo-Castrillon, F. A., Páramo-Bermúdez, G. J., & Muñoz-Betancur, J. M. (2019). Monitoring of Machining in the Cloud as a Cost Management Service and Follow of Cutting Parameters: Environment Developed With IoT Tools. *International Journal of Manufacturing, Materials, and Mechanical Engineering (IJMMME)*, 9(3), 26-41.
- [60] Lim, K. Y. H., Zheng, P., & Chen, C. H. (2019). A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing*, 1-25.
- [61] He, B., & Bai, K. J. (2020). Digital twin-based sustainable intelligent manufacturing: a review. *Advances in Manufacturing*, 1-21.



[62] Redelinghuys, A. J. H., Basson, A. H., & Kruger, K. (2019). A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing*, 1-20.

[63] Mullany, M. (2018, April 18). Percentage of Customers That Adopt An Innovation Over Time [Illustration]. *The Power of the Adoption Curve*.

<https://www.linkedin.com/pulse/basics-adoption-curve-michael-mullany/>

[64] Kuhn, T. (2019, June 19). AAS and submodels structure information in a tree of properties [Illustration]. *BaSyx Documentation*.

<https://wiki.eclipse.org/File:BaSyx.AASFormat.png>

[65] Kuhn, T. (2019, June 19). Asset Administration Shell based BaSyx architecture example [Illustration]. *BaSyx Documentation*.

<https://wiki.eclipse.org/File:BaSyx.AAS.ArchitectureExample2.png>

[66] Kuhn, T. (2018, November 29). Virtual Automation Bus example [Illustration]. *BaSyx Documentation*. <https://wiki.eclipse.org/File:BaSyx.VABExample.png>

# Anexo II

Código fuente

El código fuente de la aplicación desarrollada se encuentra en los siguientes repositorios de Github:

- <https://github.com/FIng-UdelaR/ProyGrado-2022-frontend>
- <https://github.com/FIng-UdelaR/ProyGrado-2022-backend>

Donde el primer repositorio contiene el código de la aplicación web y el segundo repositorio contiene el código de la API, el módulo de BaSyx y la herramienta de generación y ejecución de casos de pruebas.